

Dr. Bakó András

Alkalmazói softwarek

Dr. Bakó András

Alkalmazói softwarek

Felkészítő kiadó; Novotrade RT. igazgatója Rényi Gábor

Szoftver és dokumentáció másolása tilos az
39. évi III. törvény 13. paragrafusa és a 18. paragrafus 1. bekezdése alapján.

AGROÉPSZER GT. Nyomda

Tsz: 86/006

Bevezetés

A könyv megírásának célja a COMMODORE 64 gépekre meglevő gyári szoftver anyagok kezelésének könnyebbé tétele a magyar felhasználó számára. A közölt anyag a gyári kézikönyvek és az utasítások kipróbálása során nyert tapasztalatok alapján készült. A megírás során gyári kézikönyvekben levő nem tematikus tárgyalást igyekeztem tematikus rendbe szedni. Így egy paragrafusba kerültek az összetartozó utasítások is.

Az egyes szoftver anyagok leírásánál ezért alakultak ki a téma szerint összetartozó fejezetek:

Programozást segítő eljárások
Programbelövést segítő rutinok
Grafikus eljárások
Input-output utasítások stb.

Az alábbi anyagok leírását adom meg:

a.) HELP PLUS

A gyári szoftver anyag hatékony segédeszköz a programok írásához, belövé-
séhez. A SIMON'S BASIC azonban az itt levő utasítások nagy részét tar-
talmazza, és több mint kétszerannyi további utasítást, parancsot is. A rend-
szer parancsait kiptóbáltam, és mindegyik a leírásnak megfelelően működött.
Nem részleteztem az ASSEMBLER fordító használatát, mivel az a felhaszná-
lók csak kis részét érinti. Az ezen rész használatához javaslom az eredeti
leírás megfelelő részének áttanulmányozását (utolsó néhány oldal).

b.) SIMON'S BASIC

Az egyik leghatékonyabb programozást segítő szoftver anyag. A gyári leírások
(angol és német) egymástól több helyen eltérnek. Némelyik nem úgy működik,

ahogy az eredeti leírás és a mintapélda mutatja. A leírást az egyes utasítások, parancsok kipróbálása után végeztem el. A jobb megértés illusztrálására igyekeztem mindenütt példa programot is készíteni. Remélem ezek segítik az anyagok hatékony felhasználását is.

c.) SUPER GRAPHIK 64

A rendszer két változata létezik: a szalagos és a lemezes változat. A szalagos változat utasításait részletesen leírom, és mintapéldákat is közlök. A lemezes változatból eddig csak néhány érkezett hazánkba és az utasítások nagy része hasonló a szalagos változathoz. Ezért a lemezes változatból csak a többlet utasításokat írom le részletesen, és adok rá mintapéldát. Ez a többlet: a zenei utasítások és a 80x50-es grafika.

d.) C64 AIDS

A rendszer parancsai a programírást és programbelövést segítik. Tulajdonképpen a legkevesebbet tudják a 3 hasonló jellegű szoftver termékből (a, b, c)

A könyv végén összefoglalom a SIMON'S BASIC hibaüzeneteit, és itt közlöm az egyes utasítások, parancsok összefoglalását is.

Mint minden ilyen jellegű kézikönyvnél, itt is előfordulhatnak pontatlanságok, elírások. Az ilyenekért az olvasó szíves elnézését kérem. Remélem, a gép hatékonyabb kihasználását szerény írásommal elő tudtam segíteni. A további felhasználói rendszerek közlését a következő kötet fogja tartalmazni (szövegszerkesztés, adatbank szoftver stb.).

HELP PLUS

Bevezetés

A HELP PLUS egy a C64 számítógép felhasználását elősegítő utasításgyűjtemény. Az alábbi funkcionálisan elkülöníthető részekből áll:

- programírást, listázást segítő parancsok,
- programbelövést segítő parancsok,
- egyéb rendszerparancsok,
- disassembler parancsok,
- disk kezelés parancsai.

A továbbiakban végigvesszük a fenti részekhez tartozó parancsokat.

1. Programozást, listázást segítő parancsok

1.1. Automatikus sorszámkirás

A programozást segíti elő a GETLINE parancs. Ezzel a program első utasításának sorszámát (utasításcimkéjét) és az utasítás sorszámok közötti növekményt adhatjuk meg.

Alakja:

G a, b

ahol # G a GETLINE parancs rövidített alakja
a az első utasítás kezdősorszáma
b az utasítás sorszámok közötti különbség
(növekmény)

Hatása: a gép egy-egy utasítás begépelése után automatikusan kiírja a következő utasítás sorszámát

Megjegyzések:

1. A következő sorszám a RETURN gomb megnyomása után íródik ki.
2. A parancs hatása befejeződik, ha a következő sorszámot a DEL és RETURN gombokkal töröljük vagy a SHIFT és RETURN gombot egyszerre megnyomjuk.
3. Az a és b paraméterekre előírt korlátozások:
b 1 és 255 közötti értéket vehet fel,
a 63999-nél kisebb egész szám lehet.
4. Ha a és b nincs megadva, akkor
 $a = 100, \quad b = 10$

5. Ha az automatikus sorszámozás eléri a 64000 értéket, akkor tovább folyik, de az aktuális sorszámból levonja a fenti értéket.

1.2. Szabad memóriahely kérése

A programozás vagy futtatás során szükséges a még szabad memóriaterület ismerete. Ezt adja meg a * parancs.

Alakja:

*

Hatása: megegyezik a PRINT FRE (U) utasítással.

1.3. Program formázott listázása

A gépben levő program megadott részenként, oldalanként történő listázására is lehetőség van a SEITENLIST paranccsal.

Alakja:

L n

ahol #L a parancs rövidített alakja

n egy egész szám, a kezdő utasítás sorszáma.

Hatása: az n. sorszámu utasítástól kezdve kiír a képernyőre a gépben levő programból annyit, amennyi a képernyőre fér.

Megjegyzések:

1. Az n paraméter elmaradhat. Ekkor a gép a program első utasításától kezdve listáz.
2. A RETURN billentyű hatására a következő képernyőnyi programrészt írja ki.
3. Ha visszafelé, csökkenő számsorrendben akarunk listázni, akkor a billentyűt kell megvonnunk.

4. A parancs hatását a RUN/STOP billentyűvel szüntethetjük meg.

1.4. Program átsorszámozása

Egy gépben levő program utasítás sorszámainak megváltoztatását, újra sorszámozását végzi a RENUMBER parancs.

Alakja:

$\#R$ a, b

ahol $\#R$ az utasítás parancsszava,

a kezdő sorszám ($a \leq 1 \leq 63999$),

b sorszámok közötti növekmény ($1 \leq b \leq 255$).

Hatása: a gépben levő programot átsorszámozza a -tól kezdve b -esével.

Megjegyzések:

1. A paraméterek el is maradhatnak. Ilyenkor

a = 100, b = 10

értékkel dolgozik a gép.

2. Az utasítás sorszámot tartalmazó utasításokban a sorszámok automatikusan átsorszámozódnak.

Ilyen utasítások a következők:

GO TO, GOSUB, ON...,
THEN, RUN

3. Az átszámozásnál túl nagy a illetve b esetén illegális sorszámok is adódhatnak, amelyek 63999-nél nagyobbak. Ezért célszerű a $\#U$ -val leellenőrizni a programot.

1.5. Program sűritése

A felesleges üres helyek és sorszámozások összevonása a programfutást gyorsítja. Erre szolgál COMPAKTOR parancs.

Alakja:

$$\#C \ a$$

ahol $\#C$ a parancs rövidített alakja,
 a egész szám, az egy sorban elhelyezhető karakterek száma ($1 \leq a \leq 240$).

Hatása: a parancs a gépben levő program utasításait összesűriti úgy, hogy egy sorba lehetőleg a megadott a számnyi karakter legyen.

Megjegyzések:

1. A felesleges üres helyeket és megjegyzéseket a gép automatikusan kihagyja (REM-ek).
2. Azon utasításokat, ahova ugrás történik külön sorba teszi.
3. Ha van UNDEF'D STATEMENT, akkor nem hajtódik végre a sűrités.
4. Az átszámozás – a fentieket figyelembe véve – úgy történik, mint ha a

$$\#R \ 1,2$$

parancsot használtuk volna.

5. A sűritett program futása gyorsabb, de a program továbbfejlesztése nagyon problematikus a nehéz olvashatóság miatt.

1.6. Programok összemásolása

Részekben megírt programok külső tárolóról való összemásolását végzi az APPEND parancs.

Alakja:

#A "n" , d

ahol #A a parancs kulcsszava,
n a másolandó program neve,
d egységszám, ahol a másolandó program van.

Hatása: a gépben levő program végétől kezdve bemásolja a megadott d egységszámu háttértárolóról az n nevű programot.

Megjegyzések:

1. Ha d hiányzik, akkor a gép a kazettás egységet veszi háttértárolóként.
2. A másolás előtt gondoskodni kell a programrész különböző számozásáról.
3. A gépben levő gépi kódú program a másolás során felülíródik.

1.7. Egy programrészlet megkeresése

A programírás vagy programbelövés során sokszor szükséges egy karakter-sorozat (konstans, változó, kulcsszó, stb.) megkeresése. Ezt végzi a FIND parancs.

Alakja:

#F s

ahol F a parancs kulcsszava,
s a programban előforduló tetszőszerinti karaktersorozat.

Hatása: a gépben levő programban megkeresi és képernyőre kilistázza az összes utasítást, az alábbi szabályok szerint.

Szabályok:

1. Utasítás megkeresése esetén a teljes alapszót ki kell írni.
2. Függvény esetén a függvény alapszava után legalább a nyitózárrójelet ki kell írni.
3. String egy részletet a `# F" ...` paranccsal lehet megkeresni.
4. A fenti három pontban említettek kivételével tetszős szerinti programrészletet meg lehet adni.
5. Fontos megjegyezni, hogy a kereséshez az üres helynek is fontos szerepe van.
6. Az utasítások listázását a SHIFT gomb lenyomásával megállíthatjuk és továbbíthatjuk.
7. A STOP gomb lenyomása megszakítja a parancs végrehajtását.
8. A CMD megfelelő beállításával a listázás történhet például nyomtatóra is a LIST parancshoz hasonlóan.

1.8. Programsorok törlése

Egy vagy több programsor törlését végzi a DELETE parancs.

Alakja:

- a.) `# D n`
- b.) `# D -n`
- c.) `# D n-`
- d.) `# D n1 - n2`

ahol `#D` a törlés parancs,
`n, n1, n2` utasításorszámok.

Hatása: a gépben levő program

- a.) n . sorszámú sorát eltörli,
- b.) sorait eltörli egészen az n . sorszámú utasítással bezárólag,
- c.) n . sorától kezdve eltörli az utasításokat,
- d.) n_1 . sorától kezdve az n_2 . sorig kitörli az utasításokat.

Megjegyzés:

A parancs után a CONT parancsot nem használhatjuk.

2. Programbelövést segítő parancsok

2.1. Programhiba megadása megszakítás esetén

A program futásának megszakítása esetén a hiba beazonosításának munkáját segíti a HELP parancs.

Alakja:

#H

Hatása: a parancs kiadása esetén a program futása után a következő történik:

- a.) ha normálisan fejeződött be a program, akkor kiírja az utoljára végrehajtott utasítás-sort,
- b.) ha a futás megszakítása programhiba miatt történt, akkor a hibás sor kiíródik, a hibát okozó részt "reverse"-ben adja meg a gép.

Megjegyzések:

A HELP parancsot közvetlenül a program befejezése után kell kiadni.

Egyébként a parancs hatástalan.

2.2. Program futásának nyomonkövetése

Az utasítások végrehajtási sorrendjét adja meg a TRACE és a SINGLE STEP utasítás.

2.2.1. TRACE utasítás

Alakja:

#T

Hatása: a program utasításainak sorszámait végrehajtási sorrendjük szerint kiírja a képernyő legfelső sorába.

Megjegyzések:

1. A RUN, GOSUB, CONT utasítások hatására elindított program utasításai kiíródnak a képernyő 2. sorába.
2. Az utolsó 6 utasítás sorszáma a képernyő legelső sorában jelenik meg.
3. A program végrehajtása lassu, hogy a kiirt adatokat értékelni tudjuk. A SHIFT gomb lenyomásakor a program végrehajtása gyorsul.
4. Az eredmények kiírásánál figyelembe kell venni a TRACE működését, amennyiben a parancsot használni akarjuk.
5. Vigyázzunk az egyes INPUT utasításokra, mivel a TRACE ezt is módosíthatja. Ezt a kurzor üres sorba vivésével küszöbölhetjük ki. A parancs hatását a következő parancs begépelésével szüntethetjük meg:

#E

2.2.2. Programvégrehajtás soronként

A program utasításonkénti végrehajtását végzi a SINGLE STEP parancs.

Alakja:

#S

Hatása: a program utasításait egymásután végrehajtja úgy, hogy minden lépés után felfüggeszti a futást. A továbbindítás a SHIFT billentyűvel történik.

Megjegyzések:

1. A kiírás formája megegyezik a TRACE utasításnál elmondottakkal.
2. A program folyamatosan futtatható a SHIFT LOCK billentyűvel.
3. A parancs hatása megszüntethető a

#E

paranccsal

2.3. Illegális hivatkozási helyek

A nem megengedett hivatkozási helyek felülvizsgálatát végzi a #U utasítás.

Alakja:

#U

Hatása: megvizsgálja, hogy a gépben levő programban hol vannak olyan hivatkozások (GO TO, PERFORM stb), amelyekhez tartozó utasítások sorszámaikat nem tartalmazza a program. Az ilyen helyek sorszámaikat is kilistázza képernyőre.

2.4. Programváltozók kiírása

A HELP + segítségével egy program egyszerű és indexes változóinak értékeire is rákérdezhetünk.

2.4.1. Egyszerű változók értékeinek kiírása

Egyszerű változók értékeinek a kiírására szolgál a VARIABLE DUMP parancs.

Alakja:

#V

Hatása: az éppen gépben levő program egyszerű (nem indexes) változóinak tartalmát írja ki a parancs hatására.

Megjegyzések:

1. A kiírás úgy történik, hogy a képernyőn egymásután változóként külön sorba kiírja a változók értékét

$$V_i = \text{érték}_i$$

formában.

2. A kiírás lassítása szükséges, amennyiben nem férnek el egy képernyőre. Ehhez nyomjuk meg a CTRL gombot.

2.4.2. Tömbök tartalmának kiírása

Egy program tömbjei elemeinek értékét írja ki a MATRIZEN DUMP parancs.

Alakja:

#M

Hatása: Az éppen gépben levő program indexes változóinak értékét írja ki elemenként az egyszerű változókhoz hasonlóan.

2.5. Konstansok konvertálása

A memóriatartalom elemzéséhez, bizonyos memóriacímek meghatározásához sokszor szükség van a tízes és tizenhatos számrendszerbeli számok kölcsönös konvertálásához.

2.5.1. Konvertálás tizenhatosból tizesbe

Egy tizenhatos számrendszerbeli számot konvertál tizes számrendszerbeli számra a $! \#$ utasítás.

Alakja:

$$! \# \ k_1 k_2 k_3 k_4$$

ahol $! \#$ a konvertáló utasítás

k_1, k_2, k_3, k_4 tizenhatos számrendszerbeli számjegyek.

Hatása: a parancs kiadása és a RETURN gomb lenyomása után a gép a képernyőre kiírja a szám 10-es számrendszerbeli értékét.

Megjegyzések:

1. A tizenhatos számrendszerbeli szám legfeljebb 4 jegyű lehet (tehát 1, 2, és 3 jegy is szerepelhet).
2. Ha 4 jegynél többet adunk meg, az

?ILLEGAL QUANTITY ERROR IN...

hibaüzenetet kapunk.

2.5.2. Konvertálás tizesből tizenhatosba

Tizes számrendszerbeli számot tizenhatosba a $! \$$ utasítással alakíthatunk át.

Alakja:

$$! \$ \ n_1 n_2 n_3 n_4 n_5$$

ahol $! \$$

$$n_1 n_2 n_3 n_4 n_5$$

a konvertálási utasítás,

tizes számrendszerbeli számjegyek.

Megjegyzések:

1. A számjegyek száma legfeljebb 5 lehet. A legnagyobb konvertálható érték 65535 lehet tizesben.
2. Nagyobb érték megadása esetén a fenti hibaüzenet íródik ki.

3. Egyéb parancsok

3.1. Törölt program aktivizálása

A NEW paranccsal a memóriából kitörölt program visszaállítását végzi a RENEW parancs.

Alakja:

#B

Hatása: a fent elmondott.

Megjegyzések:

1. A parancs a BASIC programot megszabadítja a hozzákapcsolt gépi programtól.
2. Ha a gép valamilyen okból "lemerevedik" (azaz a kurzor eltűnik), akkor a gép aktivizálása a STOP/RESTORE gombok egyszerre történő megnyomásával lehetséges. Az esetleg elvesztett programot ilyenkor ezzel a paranccsal kaphatjuk vissza.

3.2. CMD utasítás

Az OPEN és a CMD utasítás együttes hatását eredményezi a (pa-
rancs.

Alakja:

(d

ahol (az utasítás "kulcsszava",
d egységszám (1 és 255 közötti érték).

Megjegyzések:

1. Ha a d elmarad, akkor d = 4 értéket vesz fel a gép automatikusan,

Ekkor a parancs hatása megegyezik a következő parancsok végrehajtásával:

OPEN 4,4
CMD 4

2. Ha a d értéke más, akkor az 1. pontban leírt parancsokkal egyezik meg, csak az egységszám lesz értelemszerűen más.

Az így megnyitott file lezárására szolgál a $)$ parancs.

Alakja:

tehát a parancsnak nincs paramétere.

3.3. Egységszám megváltoztatása

Egységszám megváltoztatására szolgál a f parancs.

Alakja:

$f d$

ahol f a parancs

d egységszám (4-31 közötti érték).

Hatása: HELP PLUS DOS és ASSEMBLER parancsok részére megváltoztatja a 8-as egységszámot d -re.

3.4. HELP PLUS befejezése

A HELP + "kikapcsolására" szolgál a KILL parancs.

Alakja:

$\# K$

Hatása: a HELP + kikapcsolódik, a gépben levő programot nem rontja el.

4. Disassembler parancsok

Az itt felsorolt parancsok a BASIC interpretert kikapcsolják és a gép monitor állapotban dolgozik.

4.1. Disassembler aktivizálása

A BASIC értelmező kikapcsolását és a disassembler "bekapcsolását" végzi a `]` parancs.

Alakja:

`]`

A képernyőn a következő jelenik meg alapállapotban:

0000 2F

Megjegyzés:

1. Az első négy jegy hexadecimális és a két byte-os tárcimet adja meg.
2. Az utolsó két jegy hexadecimális és a tárcimhez tartozó 1 byte hosszúságú memóriatartalmat jelenti.
3. A hexadecimális (tizenhatos számrendszerbeli) számjegyek: 0 - 9 megegyezik a tízessel. További számjegyek:

A:10, B:11, C:12, D:13

E:14, F:15

4. A disassembler 3 állapotban lehet, és minden állapotban további parancsokat ismer.

4.2. Cím és tartalom változtatása

Ebben az állapotban a címeket illetve tartalmakat változtathatjuk meg. Bekapcsoláskor ebbe az állapotba kerül a gép.

4.2.1. Cimváltoztatás

Cimváltoztató parancs a +.

Alakja:

+

A billentyű lenyomása után a 0 - F "számjegy" billentyűk valamelyikét nyomhatjuk meg. A lenyomott billentyű a négyjegyű cím utolsó jegyének helyére kerül az utolsó 3 jegy előrecsusztatása után. Az első számjegy ekkor elvész.

A további számjegy billentyűk lenyomásával a fentiekben elmondottak történnek.

4.2.2. Adattartalom változtatás

A képernyőn levő utolsó cím tartalmának megváltoztatását lehet elérni a / parancssal.

Alakja:

/

A / billentyű lenyomása után a tartalom megváltoztatása a cím megváltoztatásánál elmondott módon történik.

4.3. Listázás kijelölése

Ennek segítségével mondhatjuk meg, hogy a listázás hol történjen: printeren vagy képernyőn.

4.3.1. Listázás nyomtatón

A nyomtatón való listázási parancsa]

Alakja:

]

Hatására a disassemblálás a nyomtatóra történik.

4.3.2. Listázás képernyőn

Az utasítás a `]` , tehát megegyezik a 4.1.-el.

4.4. Memória kijelölése

A rutinokkal lehetséges a gép illetve a mágneslemez berendezés memóriatartalmának kiírása, megváltoztatása.

Gép memória kiírása

A gép memóriájának (ROM, RAM) disassemblálását végzi a `>` parancs.

Alakja:

`>`

A floppy egység memóriája (ROM, RAM) vizsgálatát írja elő a `<` parancs.

Alakja:

`<`

4.5. Listázás vezérlése

A listázás vezérlésére egy sor billentyűt használhatunk. Ezek a következők:

RETURN	a címet eggyel növeli
↑	címet eggyel csökkenti
-	a beállított címtől folytatódik a disassemblálás, ha a tartalom értelmezhetetlen * * jelenik meg
space	folyamatos disassemblálás
CTRL	kiírás lassítása
STOP	kiírás megállítása
SHIFT	kiírás átmeneti megállítása
SHIFT/LOCK	benyomva tartásáig átmenetileg áll, elengedve tovább-

4.6. További lehetőségek

4.6.1. Gépi kódu program futtatása

A * billentyűvel indíthatunk el egy gépi kódu programot. Ha a futás közben egy RST utasítást talál a gép, akkor visszatér monitor állapotba.

Ha a disk tároló van kijelölve (< parancs!), akkor akár a lemezegység tárolójában levő program is lefuttatható.

4.6.2. Memóriaterület másolása

Memóriaterületek máshova másolására szolgál a @ parancs. Ezzel mind ROM, mind RAM területről másolhatunk.

Álakja:

@

A parancs kiadása előtt be kell állítani a másolandó és a másolási terület paramétereit.

A másolandó terület megadása :

22-23 byte kezdőcím beállítása

24-25 byte végcím + 1 beállítása.

Másolás helyének megadása

"+" billentyűvel beállítjuk az első byte címét. Ezután nyomjuk meg a másolási parancsot jelentő billentyűt.

4.6.3. Disassembler kikapcsolása

A kikapcsolás az = billentyű megnyomásával történik. Ekkor a BASIC értelmező és a HELP + egyéb parancsok ismét rendelkezésünkre állnak.

5. DOS parancsok

A disk operációs rendszer parancsainak rövidítése lehetséges a HELP + segítségével. A továbbiakban ezeket ismertetjük.

5.1. Programbetöltés

a.) Alakja:

/ n

ahol n a lemezről betöltendő programnév.

Hatása: megegyezik a LOAD "n", 8 parancssal.

b.) Alakja:

% n

Hatása: megegyezik a LOAD "n", 8, 1 parancssal.

5.2. Programbetöltés és indítás

Alakja:

↑ n

Hatása: megegyezik a LOAD "n", 8 :RUN parancsokkal.

5.3. Programkiírás lemezre

Alakja:

← n

ahol n a kiírandó program neve.

Hatása: megegyezik a SAVE "n", 8 parancssal.

Hiba esetén azonban kiírja a statusz kódot és a hiba nevét is.
Névazonosság esetén az

OVERWRITE? JA/NEIN

kérdés íródik ki.

A további végrehajtás a válaszunktól függő:

J esetén a lemezen levő programot felülírja,

N esetén befejezi a parancs végrehajtását.

5.4. Kiirt program ellenőrzése

Alakja:

$\angle n$

ahol n a kiirt program neve.

Hatása: megegyezik a VERIFY "n", 8 paranccsal.

5.5. Lemeztartalom kiírása

Alakja:

$\$$

Hatása: megegyezik a LOAD " $\$$ ", 8 :LIST parancsokkal, de a gépben levő aktuális programot nem írja felül a lista.

5.6. Disk státusz kiírása

Alakja:

@

Hatása: kiírja az utolsó disk átvitel státuszát.

5.7. Parancscsatornáról olvasás

Alakja:

>

Hatása: a 15-ös parancscsatornáról egy hexadecimális jegyet beolvas.

5.8. Parancs csatorna parancsok

Az M- utasítások kivételével minden 15-ös csatornára vonatkozó parancsot kiadhatunk a @ segítségével.

A @ parancs után a PRINT # 15 után irandó paramétereket kell csak megadni. Így elmarad az OPEN, CLOSE és a PRINT parancs. Például az ALMA nevű programot az alábbi parancsokkal lehet törölni a lemezeről:

```
OPEN 15, 8, 15  
PRINT #15, "SO:ALMA"  
CLOSE 15
```

Ehelyett elegendő a következőket írni:

```
@ SO:ALMA
```

6. Assembler fordító

A **HELP PLUS** egy kétfázisú assembler fordítót is tartalmaz. A fordító a **] utasítással** aktivizálható.

Alakja:

]

A fordításhoz elő kell állítani lemezen vagy szalagon a forrásprogramot. Ezen program alakja eltér a standard assembler programok alakjától, mivel a fordító igényli az utasítások beszámozását.

A forrásprogram a következő alakú:

s (c) opc (op) (m)

ahol	s	az utasítás sorszáma (0 és 63999 közötti egész szám)
	c	nem kötelező paraméter, utasításcimke
	opc	assembler művelet kódja
	op	nem kötelező, az utasítás operandusa
	m	nem kötelező paraméter, megjegyzés.

Megjegyzések:

1. Az utasítások mindegyike külön sorban kell legyen.
2. Kötelező utasításonként: sorszám, utasítás kód.
3. A fordítás két fázisban történik:

I. fázis: címke táblázat előállítása

II. fázis: gépi kód előállítása

Ennek megfelelően a lemezről kétszer olvasódik be a forrásprogram.

SIMON'S BASIC

Bevezetés

A SIMONS' BASIC egy a C 64 BASIC értelmező kiterjesztése. Egy sor utasítást és parancsot tartalmaz a beépített értelmezőhöz képest. A parancsok legtöbbjét programba beépítve, utasításként is használhatjuk. Két verzióba létezik:

- a) Cartridge forma
- b) Floppy lemezen

Az a) esetben a rendszer aktivizálása a következőképpen történik:

- a) kikapcsoljuk a C 64 gépet, majd behelyezzük a Cartridge-t a gép hátoldali csatlakozójába,
- b) a lemezről

LOAD "x " , 8

paranccsal betöltjük a SIMONS' BASIC-et betöltő programot,
a READY üzenet megjelenése után lefuttatjuk a gépben lévő betöltőt, azaz kiadjuk a

RUN/RETURN

parancsokat.

A lemezről ezután elindul a betöltés. A betöltés után a következő üzenetet írja ki a gép fekete betűkkel fehér képernyőre:

x x x EXPANDED CBM V2 BASIC x x x
30719 BASIC BYTES FREE

Ezután rendelkezésünkre áll a több mint 100 új utasítást és parancsot tartalmazó kibővített verzió.

Megjegyzendő, hogy a rendelkezésünkre álló szabad terület mintegy 8K -val csökken. Ez az ára a kibővített lehetőségeknek.

1.1. Grafikus utasítások

A képernyőre rajzolás kényelmesebbé tétele céljából a SIMON BASIC egy sor utasítást tartalmaz. A fejezetben a rajzoló eljárásokat foglaljuk össze.

1.1.1. Rajzoló utasítások

Emlékeztetőül megjegyezzük, hogy a képernyő fekete/fehér módban 320x200, színes módban 160x200 elemi területre (cellára) van felosztva. Ezt onnan kapjuk, hogy fekete/fehérben egy karakter - hely 8x8 elemi helyre, színesben pedig 8x4 helyre van felosztva. Az egyszerűség kedvéért a továbbiakban az alábbi jelölést használjuk:

HR	fe fekete/fehér grafikus mód (HIRES)
MC	színes grafikus mód (MULTI COLOUR)

Megjegyezzük, hogy a képernyőt úgy kell elképzelni, mintha egy koordináta rendszer IV. negyedében lenne. Különbség csak annyi, hogy y értéke nem negatív.

A képernyő bal felső sarka a (0,0) pont, a jobb felső sarka a (320,0) pont, a bal alsó sarka a (0,200) pont és végül a jobb alsó sarka a (320,200) pont.

A rajzoló utasítások mindegyikében meg kell adni a rajzolási típust. Ez HR-ben és MC-ben különálló jelentéssel bír. A továbbiakban ezen típusok megadásával foglalkozunk.

HR mód

tipus	jelentése
0	kitörli a cellát
1	megjeleníti a cellát
2	az eredeti cellaállapotot ellenkezőjére változtatja
Értéke	0, ha 1 volt 1, ha 0 volt

MC mód

tipus	jelentése
0	kitörli a cellát
1	1-es színnel megjeleníti a cellát MULTI/LOW COL parancsnál
2	2-es színnel megjeleníti a cellát MULTI/LOW COL parancsnál
3	3-as színnel megjeleníti a cellát MULTI/LOW COL parancsnál
4	a cella színét inverzére változtatja a következőképpen:

0 -ról	3 -ra
1-es színről	2 -re
2-es színről	1 -re

1.1.2. Átkapcsolás grafikus módra

A HR módra való átváltás, az alap és háttérszín megadás a HIRES utasítással történik.

Alakja:

HIRES x, y

ahol: HIRES az átváltás kulcsszava,
 x a rajzolás színe (az elemi cellák színe),
 y a 8x8-as karakterhely színe (az ún. háttérszín).

A megengedett színek a C64 színei, azaz

0	fekete
1	fehér
2	piros
3	türkiz
4	bibor
5	zöld
6	kék
7	sárga
8	narancs
9	barna
10	világos piros
11	1.szürke
12	2.szürke
13	világos zöld
14	világos kék
15	3.szürke

Példa: váltsuk át a gépet HR módba kék alapszínnel és narancssárga rajzoló színnel.

Megoldás:

1.1.3. Átkapcsolás színes grafikus módra

Színes grafikus módra való átkapcsolást a MULTI utasítással érjük el.
Alakja:

MULTI x, y, z

ahol x, y, z a multikolor színek.

Az utasítás előtt mindig egy HIRES utasításnak kell állni. Az x, y, z színekre érvényes rajzolási típusnál elmondott paraméter - értéksor (lásd rajzoló utasítások paragrafus).

Példa: Legyen az első szín fehér, a második barna, a harmadik kék.

Megoldás:

HIRES 1,2:MULTI 1,9,6

Mint a fentiekben is látjuk, összesen 3 rajzolási szín adható meg a MULTI utasítással.

Ezt bővíti további 3 rajzolási szín lehetőséggel a LOW COL utasítás.

Alakja:

LOW COL p,q,r

ahol p,q,r a 0-15 értékek valamelyike.

Hatása: a MULTI-ban megadott x,y,z színek helyett a gép a továbbiakban a p,q,r színekkel dolgozik.

Az eredeti x,y,z színekre való visszatérést a HI COL utasítással tudjuk elérni.

Alakja:

HI COL

Hatása: a továbbiakban a gép a MULTI utasításban megadott színekkel dolgozik.

Megjegyzés: egy programon belül több LOW COL utasítás is kiadható, így tulajdonképpen az összes színt felhasználhatjuk rajzolásra.

1.1.4. Geometriai alakzatok megjelenítése

1.1.4.1. Egy pont megjelenítése

A PLOT grafikus utasítással lehetséges egy pont kirajzolása a képernyőn.

Alakja:

PLOT x, y, p

ahol x, y a kirajzolandó pont koordinátái,
 p a rajzolási típus.

Példa: rajzoljunk ki egy egyenest a képernyőn a PLOT utasítás felhasználásával.

Megoldás:

```
10  HIRES 1,2
20  FOR I=1 TO 320
30  PLOT I, 0.5*I,1
40  NEXT I
50  GET A$: IF A$ = "" THEN 50
```

A következő program fehér "homokszemekkel" szórja be a képernyőt:

```
10  HIRES 1, 2
20  FOR I=1 TO 200
30  FOR J=1 TO 320 STEP 10
```

```
40  PLOT RND (1) * J, RND (1) * I, 1
50  NEXT J
60  NEXT I
```

1.1.4.2. Egyenes rajzolása

Egy egyenest a kezdő és végpont koordinátáinak megadásával, valamint a rajzolási típus megadásával jeleníthetünk meg a képernyőn.

Az egyenest rajzoló utasítás

LINE x1, y1, x2, y2, p

ahol	LINE	az egyenest rajzoló utasítás kulcsszava,
	x1,y1	az egyenes kezdőpontjának koordinátái,
	x2,y2	az egyenes végpontjának koordinátái,
	p	a rajzolási típus.

Hatására az x1, y1 pontot összeköti az x2, y2 ponttal a megadott rajzolási típus szerint.

Példa: rajzoljunk néhány egyenes vonalat a képernyőre.

Megoldás:

```
10  HIRES 0,2
20  LINE 30,0,0,200,1
30  LINE 310,0,310,200,1
40  LINE 0,0,320,200,1
50  GO TO 50
```

A programot lefuttatva megkapjuk a kívánt eredményt.

1.1.4.3. Téglalap rajzolása

A REC utasítással egy téglalapot rajzoltathatunk ki a képernyőre.
Alakja:

REC x1, y1, x2, y2, p

ahol	REC	a téglalap rajzoló utasítás kulcsszava,
	x1, y1	a kirajzolandó téglalap bal felső sarkának koordinátái,
	x2	a téglalap alapjának hossza,
	y2	a téglalap magassága,
	p	rajzolási típus.

Példa: rajzoljunk ki egy négyzetet, amelynek bal felső sarkának koordinátái 10, 10, a négyzet oldala legyen 50 egységnyi, majd rajzoljunk ki egy ugyanezen kezdőponttól kiinduló 200 egység alapú és 100 egység magasságú téglalapot.

Megoldás:

```
10 HIRES 1,2
20 REC 10, 10, 50, 50, 1
30 REC 10, 10, 200, 100, 1
40 GET A$: IFA$ = "" THEN 40
```

A következő program a REC utasítás használatának egy érdekes példája:

```
10 HIRES 1,2
20 FOR I=1 TO 320 STEP 10
30 FOR J=1 TO 200 STEP 10
40 REC I,J,I,J,1
50 NEXT J
60 NEXT I
```

A kirajzolás mechanizmusát igazán multikolor módban követhetjük végig.

Ehhez az alábbi módosításokat tegyük be a fenti programunkba:

```
15  MULTI  5,4,2
```

```
40  REC  1,J,1,J,RND (1) * 4
```

1.1.4.4. Görbevonalu zárt alakzat rajzolása

Lehetőség van szabályos görbevonalu alakzat (kör, ellipszis) rajzolására is a CIRCLE utasítással.

Formája a következő:

CIRCLE x1, y1, x2, y2, p

ahol	CIRCLE	a rajzoló utasítás parancsszava,
	x1, y1	az alakzat középpontja,
	x2	az alakzat "szélessége", vízszintes "sugár"
	y2	az alakzat "magassága", függőleges "sugár"
	p	a rajzolási típus.

Mivel a képernyő nem négyzetes a rácspontok szempontjából, ezért kör rajzolása esetén az alábbi "torzitást" kell elvégeznünk:

HR mód $x2 = y2 * 1.3$

MC mód $x2 = y2 * 1.6$

Ha a kört ki akarjuk nyomtatni, úgy erre a torzitásra természetesen nincs szükség.

Példa:

```
10  HIRES 0,2
20  CIRCLE 100, 100, 100, 20, 1
30  GOTO 30
```

Egy másik alább közölt program véletlenszám generátor felhasználásával rajzol ki érdekes alakzatokat:

```
10  HIRES 2,5
20  A=INT (RND(1) *20)
30  B=INT (RND (1) *20)
40  C=INT (RND (1) *80)
50  D=INT (RND (1) * 90)
60  CIRCLE 120+A, 80+B, C, D, 1
70  GO TO 70
```

A CIRCLE utasítással kirajzolható alakzat egy részének kirajzolását végzi az ARC utasítás.

Alakja:

ARC x1, y1, a, b, c, x2, y2, p

ahol x1,y1 az alakzat középpontjának koordinátái,
a a rajzolás "kezdetének" szöge,
b a rajzolás "végének" szöge,
c a lépésköz,
x2,y2 a CIRCLE utasításnál elmondott paraméterek,
p rajzolási típus.

Hatása: x1, y1 középpontu alakzatot a szögtől a b szöögig kirajzol c lépésközzel.

Megjegyzendő, hogy a és b fokban adandó meg és az alakzat függőleges átmérője az x_1, y_1 középponttól számított egyenessel bezárt szöget kell megadni.

Példa: rajzoljunk ki egy 50,50 középpontu és 50,50 sugaru kört, majd ugyanennek a körnek "első negyedbe" eső cikkejét a 150,150 középponttal.

Megoldás:

```
10 HIRES 1,2
20 CIRCLE 50,50,50,50,1
30 ARC 150,150,0,90,1,50,50,1
40 GET A$: IF A$ = "" THEN 40
```

A lépésköz változtatására és az utasítás alaposabb megértésére futtassuk le a következő programot:

```
10 HIRES 1,2
20 FOR I=10 TO 300 STEP 10
30 ARC 150,100,0,270,1,150,100,1
40 NEXT I
50 GO TO 50
```

A CIRCLE utasítással kirajzolt alakzat hurjainak megrajzolását végzi az ANGL utasítás.

Alakja:

ANGL x_1, y_1, a, x_2, y_2, p

ahol x_1, y_1

az alakzat középpontjának koordinátái,

a

a függőleges "sugárral" bezárt szög,

Hatása: Az x_1, y_1 középpontú, x_2, y_2 tengelyű ellipszisbe egy olyan egyenes rajzol, amely α szöget zár be a függőleges sugárral.

Példa: rajzoljuk ki egy kör sugarait 30° -onként.

Megoldás:

```
10 HIRES 1,1
20 CIRCLE 160, 100, 100, 100, 1
30 FOR I=1 TO 360 STEP 30
40 ANGL 160, 100, 1, 100,100,1
50 NEXT I
60 GO TO 60
```

A fenti programot módosítsuk és futtassuk le az így kapott programot.

Cseréljük ki a következő utasításokat:

```
20 CIRCLE 160,100,160,100,1
40 ANGL 160,100,1,160,100,1
```

A multi kolor változathoz készítsük el az alábbi programot:

```
10 HIRES 1, 2
20 MULTI 2, 6, 14
30 CIRCLE 80,100,80,100,1
40 PAINT 80, 100, 3
50 FOR I=1 TO 360 STEP 10
60 ANGL 80,100,1,80,100,2
70 NEXT I
80 GO TO 80
```

1.1.4.5. Általános alakzat kirajzolása

A DRAW utasítással a "pont kurzor" mozgathatjuk vonalhúzással vagy anélkül.

Alakja:

DRAW "nnn...9", x,y,p

ahol x,y	a "kurzor" mozgató kezdete,
nn..	az alább ismertett számok valamelyike,
p	rajzoló típus.

Az n értékeit és hatását az alábbiakban összegezzük:

n értéke	művelet
0	egy cellával jobbra mozgató
1	egy cellával feljebb lépés
2	egy cellával lejjebb lépés
3	egy cellával balra lépés
4	meg egyezik a 2-vel
5	egy cellával jobbra mozgató rajzolóval
6	egy cellával feljebb mozgató rajzolóval
7	egy cellával lejjebb mozgató rajzolóval
8	egy cellával balra mozgató rajzolóval
9	mozgató befejezése.

Hatása: x, y koordinátáktól kezdve az idézőjelben megadott számoknak megfelelő mozgásokat (és rajzolásokat, ha vannak) hajtja végre.

Megjegyzések:

1. Az 1-4 számok csak mozgatják a "kurzort", míg az 5-8 rajzolnak is a képernyőre.
2. A DRAW utasítás csak a ROT utasítás után rajzol ki a képernyőre.

A DRAW paranccsal létrehozott alakzatokat a ROT utasítással jeleníthetjük meg.

Alakja:

ROT s, m

ahol s 1 és 8 közötti érték, a forgatás szöge,
 m az eredeti méret növelési faktora, 1 és 255 közötti érték.

Hatása: a DRAW utasítással megadott alakzatot az s értéknek megadott szögben m -szeresére növelve kirajzolja.

A forgatási szöget az alakzat eredeti helyzetéhez mérten kell érteni.

A szög (s) értékei:

s	forgatási szög
0	nincs forgatás
1	45°
2	90°
3	135°
4	180°
5	225°
6	270°
7	315°

Példa: Rajzoljunk ki egy A betűt az 50,50 ponttól kezdve, majd forgassuk meg növekvő méretekkel a 100, 100 pont körül.

Megoldás:

```

10  HIRES 1,0
20  B$ = "6666666666666666555555"
      15      6
30  C$ = "7777777777777777111111118888889"
      15      9      6
40  ROT 0,1
50  DRAW B$+C$, 50,50,1
60  FOR I=0 TO 7
70  ROT I,0.5*I+2
80  DRAW B$+C$, 100, 100,1
90  NEXT I
100 GET A$:IF A$="" THEN GO TO 100

```

Ha lépésenként akarjuk végignézni a forgatást, akkor tegyük be a programba a következő utasítást:

```
85  GET A$:IF A$="" THEN 85
```

Ha a fenti programot színesben is meg akarjuk csinálni, javasoljuk a következő módosítások elvégzését:

```
15  MULTI 6,8,10
```

```
80  DRAW B$+C$, 100, 100, (RND(1)*3)+1
```

1.1.5. Alakzatok színezése

1.1.5.1. Zárt alakzat színezése

Egy zárt alakzat kiszínezése megoldható a SIMON BASIC-ban a PAINT utasítással.

Ennek formája a következő:

PAINT x,y,p

ahol PAINT	az utasítás kulcsszava,
x,y	az alakzat tetszésszerű pontjának koordinátái,
p	rajzolási típus.

Végrehajtása: a rajzolási típusban megadott színre kifesti a megadott alakzatot. Ha az alakzat nem volt zárt, akkor az egész képernyőt a megadott színre "festi" át.

Példa: színezzünk ki egy ellipszist.

Megoldás:

```
10  HIRES 3,2
20  CIRCLE 110, 100, 130, 15,1
30  PAINT 110, 100, 1
40  GO TO 40
```

Az alakzatok színezésére és a színskála tanulmányozásához futtassuk le az alábbi programot:

```
10  HIRES 0,0
20  MULTI 1,1,1
30  FOR I=0 TO 15
```

```
40  REC 10*1, 0, 10, 200, 1
50  NEXT I
60  FOR K=0 TO 15
70  LOW COL K,K,K
80  PAINT 10*K+3, 100,1
90  NEXT K
100 GET A$:IF A$="" THEN GO TO 100
```

1.1.5.2. Színes téglalap rajzolása

Egy téglalapot a REC utasítással rajzolhatunk ki. Ezt színeezni a PAINT utasítással lehet. A két funkciót egyszerre és gyorsabban látja el a BLOCK utasítás.

Alakja:

BLOCK x1, y1, x2, y2, p

ahol

x1,y1,x2,y2 a REC utasításnál elmondott paraméterek,
p rajzolási típus.

Hatása: kirajzol egy téglalapot az x1, y1, x2, y2 koordináták figyelembevételével a REC utasításnál elmondottak szerint.

Amennyiben multi-color módot használunk, úgy a p az ott elmondott értékeket veheti fel.

Példa:

1. Rajzoljunk ki két négyzetet a képernyő jobb felső sarkában az utasításokban megadott méreteken.

Megoldás:

```
10  HIRES 1,0
20  BLOCK 1, 1, 100, 100, 1
30  BLOCK 50, 60, 80, 100, 2
40  GET A$:IF A$="" THEN GO TO 40
```

Eredmény: képernyő bal felső sarkában egy fehér négyzet, majd egy fekete négyzet jelenik meg.

2. A fenti programot multi-color módban is elkészíthetjük:

```
10  HIRES 1,0:MULTI 6,8,10
20  BLOCK 1,1,100,100,1
30  BLOCK 50, 60, 80, 100, 2
40  GET A$:IF A$="" THEN GO TO 40
```

Eredmény: Kék négyzeten belül sárga kis négyzet.

1.1.6. Rajzok feliratozása

1.1.6.1. Karakterek megjelenítése

A grafikus módban is szükséges bizonyos jelek, szövegek képernyőre írása. Egy karakter kiírását végzi el a CHAR utasítás.

Alakja:

CHAR x, y, c, p, s

ahol CHAR	a karakter rajzolás kulcsszava,
x,y	karakter kirajzolásának koordinátái,
c	a kirajzolandó karakter POKE kódja,
p	a rajzolási típus,
s	a kirajzolandó karakter mérete.

Hatása: a megadott pontba kirajzolja a POKE kóddal megadott karaktert a rajzolási típus szerint. Az s méret értéke 1, akkor normál méretben rajzolja, ha más, megfelelő méretalakítás történik.

Példa:

```
10  HIRES 12, 9
```

```
30  X=INT (RND(1) * 15)
40  CHAR T * 10, 80, T, 1, X
50  NEXT
60  GO TO 60
```

1.1.6.2. Karakterláncok kiíratása

Egyszerre több karaktert is megjelenítő utasítás a TEXT.

Alakja:

TEXT x,y, "(t) b", p, s, i

ahol	TEXT	a karaktersort megjelenítő utasítás kulcsszava,
	x,y	az első karakter koordinátái (szöveg kezdete!)
	t	az alsó illetve felső állást megadó paraméter,
	b	a kiírandó karaktersorozat,
	p	rajzolási típus,
	s	a karakterek mérete,
	i	üres cellák száma a karakterek kezdete között.

A t értékei a következőképpen adhatók meg:

CTRL-a billentyű: nagybetűk

CTRL-b billentyű: kisbetűk.

Hatása: a megadott (x,y) ponttól kezdve megfelelő állásban vízszintesen kiírja a b szöveget a rajzolási típusnak és a méretnek megfelelően.

A méretre vonatkozó előírásokat a CHAR utasításnál mondtuk el.

Példa:

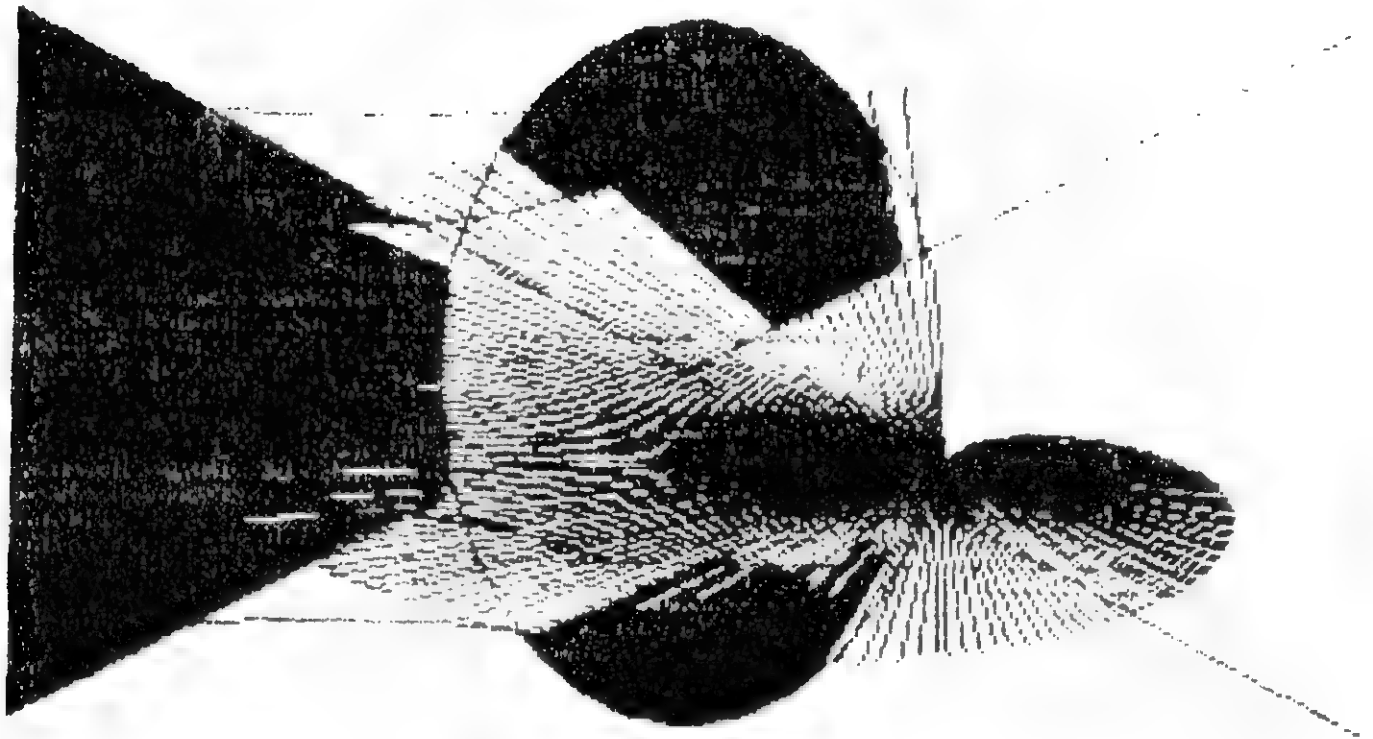
```
10  HIRES 6,4
20  TEXT 60, 30, "B IROK IS", 1,3,10
30  TEXT 60,60 "UGYE JO?", 1,5,20
40  TEXT 60,100 "A ALMAT ESZEM", 1,8,25
50  GO TO 50
```

1.1.6.3. Mintaprogram

Az alábbiakban több utasítást használó programot közlünk:

```
10  HIRES 2, 5,
20  LINE 0, 0, 320, 200, 1
30  LINE 320, 0, 0, 200, 1
40  CIRCLE 160, 100, 60, 100, 1
50  ARC 80, 0, 152, 207, 1, 100, 30, 1
60  ARC 80, 200, 333, 28, 1, 100, 30, 1
70  ARC 80, 0, 142, 217, 1, 100, 60, 1
80  ARC 80, 200, 323, 38, 1, 100, 60, 1
90  FORT=5TO358STEP2
100 ANGL 220, 125, T, T/1.5, T/3.3, 1
110 NEXT
120 PAINT1, 1, 1
130 PAINT 200, 100, 1
140 PAINT 160, 20, 1
150 PAINT 160, 180, 1
160 COPY
170 GO TO 170
```

A fenti programhoz tartozó rajzot a nyomtatón készítettük el:



1.1.7. További grafikus utasítások

1.1.7.1. CSET utasítás

A CSET utasítással lehetőség van a karakterkészletek váltására és a legutolsó grafikus képernyő visszarajzolására is.

Alakja:

CSET i

ahol i 0 és 2 közötti érték

Hatása:

- $i = 0$ esetben a gép felső állása/grafikus karakterkészletét választja
- $i = 1$ esetben a felső/alsó állás karakterkészletét választja
- $i = 2$ esetén átvált grafikus üzemmódba és ki-rajzolja a képernyőre a grafikus képernyő utolsó rajzát.

Megjegyzés: ha az utolsó rajz színes üzemmódban jelent meg, akkor a CSET 2 utasítást egy MULTI utasításnak kell követni.

1.1.7.2. Egy pont állapotának lekérdezése

Sokszor szükséges a grafikus képernyőn egy "pontról" eldönteni, hogy történt-e oda rajzolás vagy sem. Ezt végzi el a TEST függvény.

Alakja:

TEST (x, y)

ahol x, y a keresett pont paramétere

Hatása:

- a TEST numerikus értékű függvény értéke 1, ha a pontban történt rajzolás,
- ha a "pont üres" akkor a TEST értéke 0.

1.2. Hagyományos utasítások kiterjesztése

1.2.1. Feltételes utasítás kiterjesztése

1.2.1.1. IF ... THEN ... ELSE

A BASIC-ban megadott IF...THEN szerkezet mellett az IF...THEN...ELSE kifejezést használhatjuk.

Az utasítás alakja:

$$\text{IF } L \text{ THEN } u_1[:u_2:\dots:u_n] : \text{ELSE}:v_1[:v_2:\dots:v_m]$$

ahol L logikai kifejezés (reláció),

u_1, v_1 végrehajtható BASIC utasítások.

Végrehajtása: ha L igaz, az $u_1(u_2, \dots, u_n)$ utasítások hajródnak végre, egyébként a $v_1(v_2, \dots, v_m)$ utasítások.

Példa:

$$\text{IF } 5 < a + b \text{ THEN } X = 50 : \text{ELSE} : X = 100$$

Ha $5 < a + b$, akkor X értéke 50, egyébként X értéke 100.

1.2.1.2. RCOMP utasítás

Az RCOMP utasítás formája:

$$\text{RCOMP}:u_1[:u_2:\dots:u_n] : \text{ELSE}:v_1[:v_2:\dots:v_n]$$

ahol RCOMP, ELSE az utasítás kulcsszavai,

u_i, v_i végrehajtható utasítások.

Végrehajtása: ha az RCOMP utasítást megelőző IF utasításban a feltétel értéke igaz, akkor az u_i , egyébként u_{i+1} utasítások hajtódnak végre.

Példán

```
10 IF X+Y = Z THEN P = 20:ELSE:P=30
20 RCOMP : R = 15 : ELSE : R = 1000.25
```

Eredmény: ha $X + Y = Z$, akkor $P = 20$, $R = 15$, máskülönben $P = 30$, $R = 1000.25$

1.2.2. Ciklusutasítás kiterjesztése

A FOR ciklusutasítás mellett egy sor más ciklusutasítást is használhatunk a SIMON BASIC-ban.

1.2.2.1. REPEAT ... UNTIL utasítás

Ennek formája:

```
c1 REPEAT
c2 u1
... ..
cn un-1
cn+1 UNTIL L
```

ahol REPEAT, UNTIL a ciklusutasítás címszavai

c_1	utasítások számai ($i=1, 2, \dots, n+1$),
L	logikai kifejezés (reláció),
u_i	utasításmozgat.

Végrehajtása: a $c_2 \dots c_n$ utasítások addig kerülnek végrehajtásra, ameddig az L kifejezés értéke igaz nem lesz.

Példa:

```
100 REPEAT
110 INPUT X
120 UNTIL X = 50
```

A fenti utasítássorozat addig kerül végrehajtásra, ameddig az 50-es értéket be nem olvastuk az X-be.

1.2.2.2. LOOP ... EXIT IF ... END LOOP

Az utasítás formája a következő:

```

c1    LOOP
c2    u1
.
.
ci    EXIT IF L
.
.
cn+1  END LOOP
```

ahol LOOP, EXIT IF, END LOOP az utasítás kulcsszavai

c_i az utasítások számai (i=1,2, ...,n),

L logikai kifejezés (reláció),

u_i utasítássorozat.

Végrehajtása: a c₂...c_n utasítások addig hajtódnak végre, ameddig az L logikai kifejezés igaz nem lesz.

Példa:

```
100  LOOP
110  INPUT X
120  EXIT IF X = 50
130  END LOOP
```

A fenti utasítássorozat addig hajtódik végre, ameddig az input adatok között az 50-es elő nem fordul.

Megjegyzés: a c_i utasítássorozatban az EXIT IF utasítás többször is előfordulhat.

1.2.3. Zárt szubrutin

Szubrutint a GOSUB utasítás mellett a PROC utasítással is létrehozhatunk. Az ilyen szubrutin ún. zárt szubrutin lesz.

Formája:

```
 $c_1$     PROC n
 $c_2$      $u_1$ 
.
.
 $c_n$      $u_{n-1}$ 
 $c_{n+1}$   END PROC
```

ahol PROC eljárásfej,
 n eljárásnév,
 c_i $i=2, \dots, n$ szubrutin törzsét alkotó eljárások,
 END PROC szubrutin végét jelző utasítás.

A szubrutint a következő utasítással lehet végrehajtani:

EXEC n

ahol EXEC a szubrutint hívó utasítás kulcsszava,
 n a kiírandó szubrutin neve.

Végrehajtása: az n neveű szubrutint végrehajtja, majd az EXEC utasítást követő utasítás után folytatódik a program végrehajtása.

Példa:

```
100  EXEC ABC
110  PRINT "SZUBRUTIN VÉG"
120  . . .
300  PROC ABC
310  FOR I=1 TO 100 : PRINT I, I*I
320  NEXT I
330  END PROC
```

A program végrehajtása során az EXEC aktivizálja az ABC nevű szubrutint. Így a képernyőre 1-től 100-ig ezen számok és a négyzeteik kerülnek kiírásra. Ezután a SZUBRUTINVEG kiírással folytatódik a program végrehajtása (110-es utasítás!).

1.2.4. Cimkére ugrás

A PROC utasítást a zárt szubrutin létrehozása mellett ugró utasítás egy másik formájának alkalmazásához is használhatjuk.

Formája:

c PROC n

ahol c utasításcímke,
PROC ugrás helyét megadó parancs,
n szimbolikus név.

Végrehajtása: a c címkétől kezdődő utasítássorozatot a CALL n utasítással aktivizálhatjuk. Így a CALL n végrehajtás szempontjából megegyezik egy GO TO c utasítással. A különbség csak az, hogy az n egy név, nem pedig egy címke.

Jegyezzük meg: zárt szubrutinra nem alkalmazhatjuk a CALL utasítást.

Példa:

```
.....  
100 CALL ALMA  
110 X = 25  
  
.....  
200 PROC ALMA  
210 X = 50  
220 END PROC
```

A fenti utasítássorozat hatására először X az 50, majd a 25 értéket veszi fel.

1.2.5. Helyi és globális változók

Sokszor szükséges a programírás során azonos változókat más célra használni. Különösen fontos ez, ha többen dolgoznak egy program kifejlesztésén és azonos változókat használnak. Valamely programrészben (különösen szubrutinoknál) egy változó értékének elmentését, és új változóként való használatát a `LOCAL` utasítással tudjuk megoldani. Mindaddig érvényes az utasítás, ameddig egy `GLOBAL` parancsot ki nem adunk, aminek hatására a változó aktuális értékei a `LOCAL` parancs előtti értékek lesznek.

A parancsok formái:

```
c  LOCAL  v1, v2, ..., vn
c  GLOBAL w1, w2, ..., wm
```

ahol `c` utasításcímke,
`LOCAL`, `GLOBAL` utasítás kulcsszók,
`vi`, `wi` változó nevek.

Végrehajtásuk: a `LOCAL` parancs hatására a `v1, v2, ..., vn` változók értéke eltárolódik és értékük nulla lesz.

Ezen változók értéke a rájuk vonatkozó utasításig a helyi használat eredményeképpen kapott érték lesz. A `GLOBAL` utasítás után a `w1` listában szereplő változók visszakapják a rájuk vonatkozó `LOCAL` parancs előtti értéküket.

Példa:

```
. . . . .  
100  X=425.2: Y$="ALMA" :Z%=11111  
110  PRINT X,Y$,Z%  
120  LOCAL X,Y$  
130  PRINT X,Y$,Z%  
140  X=1.45:Y$="HOLVOLT"  
150  PRINT X,Y$,Z%  
160  GLOBAL Y$:LOCAL Z%:Z%=22222  
170  PRINT X,Y$,Z%  
180  GLOBAL X  
190  PRINT X,Y$,Z%
```

A fenti utasítások hatására a következők jelennek meg a képernyőn:

```
425.2  ALMA      11111  
0      11111      0  
1.45   HOLVOLT 11111  
1.45   ALMA      22222  
425.2  ALMA      22222
```

1.2.6. Számitott ugró utasítás

Aritmetikai kifejezés értékétől függő utasítás a CGOTO.

Alakja:

, CGOTO e

ahol e aritmetikai kifejezés.

Hatása: GO TO c utasítással egyezik meg, ahol c az e kifejezés értéke.

Megjegyzés: Amennyiben e valós, akkor a c az e értékének

1.2.7. DATA pointer állítása

A READ utasítás a BASIC-ben úgy hajtodik végre, hogy beolvassa a DATA ak' elis p- i- ciójától a kívánt adatokat. Az aktuális pozíció megváltoztatására szolgál a RESET utasítás.

Alakja:

RESET c

ahol c egy DATA utasítás címkéje.

Hatása: a következő READ utasítás a c címkével megadott DATA utasításra vonatkozik.

Példa:

```
10 INPUT N$
20 IF N$ = "A" THEN RESET 80:GO TO 40
30 RESET 90
40 FOR I=1 TO 10
50 READ A:PRINT A
60 NEXT I
70 STOP
80 DATA 1,2,3,4,5,6,7,8,9,10
90 DATA 10,20,30,40,50,60,70,80,90,1,2
```

A program végrehajtása során

- ha az A billentyűt nyomjuk meg, akkor a gép kiírja az 1,2,...,9,10 adatokat
- ha a B billentyűt nyomjuk meg, akkor a 10,20,...,90,1 adatokat.

1.2.8. Olvasással-írással kapcsolatos műveletek

1.2.8.1. Szöveg kiírás közére

Képernyőre szövegek elhelyezésére szolgáló utasítás a CENTRE.
Alakja:

CENTRE "s"

ahol s a képernyőre elhelyezendő karakterlánc.

Hatása: az s karakterláncot elhelyezi a képernyő aktuális sorának közepére.

Példa: A képernyő aktuális sorának közepére írassuk ki a

VEGEREDMENY

szöveget.

CENTRE "VEGEREDMÉNY"

1.2.8.2. Kiíratási kép szerkesztése

Numerikus adatok kiíratási képének meghatározására szolgál a USE...PRINT utasítás.

Alakja:

- a) USE "p.q",v:PRINT
- b) USE "s1 p.q s2",v:PRINT

ahol p,q # jelek sorozata,
s1,s2 karakterláncok,
v karakter típusu változó.

Hatása:

- a) a v változóban levő numerikus értéket p egészre és q tizedesre kiírja a képernyőre.

- b). A α változóban levő numerikus értéket az az a)-ban leírtak szerint kiírja, de előtte az s1, utána az s2 szöveget is megjeleníti.

Példa: Tekintsük a következő programot

```
10 INPUT A
20 A$=STR$(A)
30 USE "EGESZ:          TIZEDES",A$:PRINT
40 GO TO 10
```

A fenti programot futtassuk a következő 4 értékkel

0.5
125.589
2.41
1125.5643

A képernyőre a gép a következőket írja ki:

```
EGESZ:   .5 TIZEDES
. EGESZ: 125.58TIZEDES
EGESZ:  2.41TIZEDES
EGESZ: 125.56TIZEDES
```

Megjegyzendő, hogy a kiíratás során – hibajelzés nélkül – kezdő és végszámjegyek tűnhetnek el, mint az az utolsó érték kiíratásánál is előfordult. Ezért a USE utasítás használata esetén ügyeljünk a számok nagyságrendjére.

1.2.8.3. Kiíratás helyének megadása

Szövegek adott sor és oszlop kezdettel történő kiíratását végzi a PRINT AT utasítás.

Alakja:

a) PRINT AT(p,q)"s"

b) PRINT AT(p,q)v

ahol s karakterlánc,

v változó,

p,q oszlop illetve sorszám.

Hatása: a) a p. oszlop q. sorától kezdve kiírja az s szöveget

b) a p. oszlop q. sorától kezdve kiírja a v változó értékét.

Megjegyzendő, hogy több AT összetétel is előfordulhat egy PRINT utasításban.

Példa: Irassuk ki az OSSZESEN szöveget a 3. sor 15. oszlopától az ALAPBER szöveget az 5. sor 10. oszlopától és a POTLEKOK OSSZEGE szöveget ugyanezen sor 20. pozíciójától.

Megoldás:

```
10 PRINT AT(15,3) "OSSZESEN" AT(10,5)
   "ALAPBER"
```

```
20 PRINT AT(20,5)"POTLEKOK OSSZEGE"
```

Megjegyzendő, hogy a sorok 0-tól 24-ig, az oszlopok 0-tól 39-ig számozódnak.

1.2.8.4. Bemenő adatok ellenőrzése

A bemenő adatok beolvasására, típusának és maximális hosszának ellenőrzésére használatos a `FETCH` utasítás.

Alakja:

`FETCH "v", n, vn`

ahol `v` az utasítás vezérlő karaktere,

`n` a beolvasandó adat maximális hossza karakterekben,

`vn` a változó neve, ahova a beolvasás történik.

Hatása: a `vn` változóba beolvas a billentyűzetről legfeljebb `n` karaktert a `v` vezérlő karakternek megfelelően.

A `v` értékei a következők lehetnek:

karakter értéke	megengedett input karakter
<code>CLR/HOME</code>	nem siftelt alfabetikus karakter
<code>CURSOR DOWN</code>	numerikus karakter
<code>CURSOR LEFT</code>	alfabetikus és siftelt karakter

Ha a karakterlánc típusa nem megfelelő, akkor a futás nem folytatódik, amíg megfelelő típusu adatot nem kapott. Megengedett hosszabb lánc esetén csak az első `n` számot fogadja el.

Példa:

```
10  FETCH "CLEAR/HOME",2,A$
20  PRINT "TOVABB"
. . . . .
```

A fenti program csak akkor indul tovább, ha egy, vagy két betűt leütünk és megnyomjuk a `RETURN` gombot.

1.2.8.5. Funkció billentyű használata

Az F1-F8. billentyűk egyik felhasználását segíti az INKEY függvény.

Alakja:

INKEY

Hatása: - az INKEY értéke nulla, ha egyik billentyűt sem nyomtuk le,

- és 0, ..., 16 ha F1, ..., F4 és esetleg a SHIFT vagy COMM. billentyű valamelyikét nyomjuk le

Példa: Tekintsük az alábbi programot:

```
10  A=INKEY
20  IF A=0 THEN 10
30  PRINT A
40  ON A GO TO
      100,200,300,400,500,600,700,800
. . . . .
```

A fenti program addig "olvassa" a funkció billentyűket, ameddig valamelyiket le nem nyomtuk. Valamely funkció billentyű lenyomása után a megfelelő százzal kezdődő címkeire ugrik.

Példa: tekintsük a következő programrészt:

```
10  PRINT "NYOMJA MEG AZ F1 BILLENTYUT"
20  INKEY
30  IF B=1 THEN 60
40  PRINT "NEM F1-ET NYOMOTT"
50  GO TO 10
```

A fenti programot futtatva a végrehajtás csak akkor folytatódik a 60-as utasításnál, ha az F1 billentyűt megnyomtuk.

Megjegyzés:

Az INKEY értékei

Billentyű nyomás	INKEY
F1	1
SHIFT, F1	2
F3	3
SHIFT, F3	4
F5	5
SHIFT, F5	6
F7	7
SHIFT, F7	8

1.3. Egyéb utasítások

A paragrafusban egy sor hasznos utasítást mutatunk be. Ezek közül kiemeljük az egyik leghasznosabbat, amellyel az előforduló hibákat a programból "kijavíthatjuk" és a futás így tovább folytatható. Az utasítás megfelelőjét a PL/I-ből a nyelvet ismerők már használták.

A programok titkosításával az illetéktelen továbbfejlesztők, programrészeket egyéb célokra felhasználók munkáját lehet megnehezíteni.

1.3.1. Programfutas felfüggesztése

A program futásának megadott időre történő felfüggesztését végzi a PAUSE utasítás.

Alakja:

- a) PAUSE "u", i
- b) PAUSE i

ahol u tetszőszerinti szövegkonstans,
i másodpercek száma.

Hatása: a) A PAUSE utasításhoz érve a futást felfüggeszti i másodpercre és kiírja az idézőjelbe írt szövegkonstanst.
b) A futás felfüggesztődik i másodpercre.

A program végrehajtása mindkét esetben automatikusan folytatódik i másodperc múlva. Ha az i másodpercet nem akarjuk kivágni, akkor a RETURN billentyűvel továbbíthatjuk.

Példa:

```
10 PAUSE "ADD MEG AZ A-t", 30
20 INPUT A
. . . . .
```

A fenti programrész végrehajtása során a gép kiírja az ADD MEG AZ A-t szöveget, 30 másodpercig áll, majd végrehajtja a 20-as utasítást.

1.3.2. Hibák lekezelése programból

A futás közben előforduló hibák egy részét lekezelhetjük programból is, így ezek miatt nem szakad meg a program futása.

A következő hibák lekezelése lehetséges így:

- 1 túl sok file
- 2 állomány már meg van nyitva
- 3 állomány nincs nyitva

- 4 állományt nem találja
- 5 egység nem áll rendelkezésre
- 10 NEXT FOR nélkül
- 11 szintaktikus hiba
- 12 RETURN GOSUB nélkül
- 13 DATA utasításban már nincs adat
- 14 nem megengedett mennyiség
- 15 tulcsordulás
- 16 memóriát túlhaladta
- 17 ilyen utasításcímke nem létezik
- 18 nem megfelelő tömbelem megadás
- 19 tömböt újra definiálunk
- 20 nullával való osztás
- 21 csak direkt módban használható
- 22 típus keveredés
- 23 túl hosszú karaktertípusú konstans

A fenti hibák esetén a hibaanalizálást a következő változók segítik elő:

ERRLN	hibás utasítás sorszáma
ERRN	hiba kódja

A hiba kezelését a következő utasítással lehet megoldani:

ON ERROR:GO TO c

ahol ON ERROR a hibakezelés kulcsszava,
c utasításcímke, amelyre a hiba esetén ugrik.

A hiba lekezelését a fenti két (ERRLN, FRRN) változók felhasználásával végezhetjük el.

A tennivalók elvégzése után a

NO ERROR

paranccsal továbbindíthatjuk a futást.

Az OUT parancs adja vissza a vezérlést a COMMODORE-64 BASIC hiba kezelő rutinjának.

1.3.3. Programok titkosítása

Egy program mások által történő átalakításának meggátolására a program megjelölt sorait el lehet "tüntetni". Az eltüntetendő sorok megjelölésére szolgál a DISAPA utasítás, sorok eltüntetését végzi a SECURE parancs.

A DISAPA utasítás formája:

DISAPA:

Ezt a formát kell beütni azon utasítások sorszáma után, amelyeket el akarunk tüntetni.

Példa: Tekintsük az alábbi programot:

```
10 DIM A(100,20)
20 FOR I=1 TO 100
30 FOR J=1 TO 5
40 IF I=J THEN A(I,J) = 999
50 NEXT J
60 NEXT I
```

Ha a program 20. és 40. sorszáma utasításait el akarjuk tüntetni, akkor megjelöljük a DISAPA utasítással, azaz kicseréljük a következőkre:

```
20  DISAPA:FOR I=1 TO 100
40  DISAPA:IF I=J THEN A(I,J) =999
```

Ezután tüntessük el a kijelölt utasításokat, azaz adjuk ki a SECURE 0 parancsot:

```
SECURE 0/RETURN
```

Listázzuk ki a programot (LIST RETURN).

Az eredmény:

```
10  DIM A(100,20)
20
30  FOR J=1 TO 20
40
50  NEXT J
60  NEXT I
```

2. BASIC függvények kiterjesztése

2.1. Matematikai függvények

2.1.1. Modulus függvény

A függvény formája a következő:

$$\text{MOD } (a,b)$$

ahol MOD a modulus függvény kulcsszava,
a,b aritmetikai kifejezés.

Értelmezése: a függvény értéke a modulo b.

2.1.2. Egészrész

A függvény formája a következő:

$$\text{DIV } (a,b)$$

ahol DIV az egész osztás függvény kulcsszava,
a,b aritmetikai kifejezés.

Értelmezése: a függvény értéke az a legnagyobb egész szám, amelyet b-vel megszorozva a-tól kisebb vagy egyenlő számot kapunk.

Példa: Tekintsük a következő programot:

```
10 INPUT A,B
20 PRINT A, ":", B, "=", DIV (A,B)
30 PRINT "A MARADEK", MOD(A,B)
```

Lefuttatva a következők íródnak ki a képernyőre:

A	B	KIÍRÁS
21	6	A:B = 3 A MARADEK 3
16	2	A:B = 8 A MARADEK 0

2.1.3. Tizedesrész függvény

A függvény formája:

FRAC (k)

ahol FRAC a függvény kulcsszava,

k aritmetikai kifejezés (konstans, változó...).

Végrehajtása: A függvény értéke a k kifejezés tizedes része.

Példa: a) PRINT FRAC (3.456), FRAC (1/2)

A képernyőn a következő számok íródnak ki: .456, .5

b) X = FRAC (12/5)

Az X változó értéke 0.4 lesz.

2.2. Logikai műveletek kiterjesztése

A BASIC-ben meglevő AND, OR és NOT logikai operátorok mellett a kizáró vagy műveletet is használhatjuk a SIMON BASIC-ben.

A logikai függvény alakja:

EXOR (a, b)

ahol EXOR a kizáró vagy függvény kulcsszava,
a,b egész (vagy valós) számok.

Végrehajtása: az a és b számokat bináris számmá alakítja,
számjegyenként végignézi, ha a két számjegy azo-
nos, akkor az eredmény 0, egyébként 1.

Példa: a = %00101101
b = %00011100

eredmény 00110001 ezért a 49 érték íródik ki a

PRINT EXOR (%00101101, %00110001)

hatására.

2.3. Karakter műveletek

2.3.1. Beszurás karaktersorozatba

Sokszor szükséges egy karaktersorozatba egy másikat beszurni. Ezt végzi el az INSERT függvény.

Alakja:

INSERT ("s1", "s2", n)

ahol s1,s2 karakterláncok,
s1 a beszurandó karakterlánc,
s2 a karakterlánc, amelybe a beszurás történik,
n azon pozíció helye, ahova a beszurás történik s2-ben.

Megjegyzendő, hogy s₁ és s₂ karaktereinek együttes száma nem ha-
ladhatja meg 255-öt.

Példa: Tekintsük a következő programot

```
10  A$ = "ALMA  ALATT"  
20  B$ = INSERT ("A  FA", A$, 5)  
30  C$ = INSERT ("MEG  EGY", "EGY  KETTO", 4)  
40  PRINT B$  
50  PRINT C$
```

A program lefuttatása után a következő szövegeket írja ki a gép a képernyőre

```
ALMA A FA ALATT  
EGY MEG EGY KETTO
```

A függvényt karakterláncok összehasonlítása esetén is használhatjuk.

Az összehasonlítás baloldalán egy numerikus típusú változó van, amelynek értéke -1, ha az összehasonlítás igaz, egyébként 0.

Példa: Tekintsük a következő programot:

```
10  A = INSERT ("-OTT", "ITT", 3) = "ITT-OTT"  
20  PRINT A
```

A futtatás eredményeképpen a következő érték jelenik meg a képernyőn:

```
-1
```

A függvénnyel kapcsolatos hibaüzenetet a Függelékben írjuk le.

2.3.2. Karakterlánc egy részének kicserélése

Egy karakterlánc végének a megváltoztatására szolgál az INST függvény.

Alakja:

INST ("s1", "s2", n)

ahol s1, s2 karakterláncok,
 s2 a karakterlánc, aminek a végét változtatjuk meg,
 s1 az s2 karakterlánc végére irandó karakterlánc,
 n azon pozíció helye az s2-ben, amelytől kezdve
 az s1 láncot helyettesítjük.

Az s1 és s2 akár melyike lehet karaktertípusú változó is.

Példa: Tekintsük a következő programrészletet:

```
10  A$ = INST ("ALMA", "PIROS-KEK", 5)
20  PRINT A$
```

Lefuttatva a következőket írja ki a gép:

PIROSALMA

2.3.3. Karakterlánc helyének megkeresése

Egy karaktersorozatban egy részsorozat megkeresésére és a helyének megállapítására szolgál a PLACE függvény.

Alakja:

PLACE ("s1", "s2")

ahol s1, s2 karakterláncok,
 s1 a karakterlánc, amit keresünk,
 s2 az a karakterlánc, amelyben az s1-et keressük.

A függvény értéke az s1 karakterlánc első pozíciójának a helye s2-ben, ha s1 része s2-nek. Ellenkező esetben a függvény értéke nulla.

Példa: Tekintsük az alábbi programrészletet

```
10 A$ = "ALMA A FA ALATT"  
20 X = PLACE ("FA", A$)  
30 Y = PLACE ("ALA", A$) : Z = PLACE ("N", A$)  
40 PRINT X, Y, Z
```

A program végrehajtása során az alábbi értékek íródnak ki a képernyőre:

8 11 0

2.3.4. Karakterlánc többszörözése

Karakterlánc többszörözésére szolgáló eljárás a DUP.

Alakja:

DUP("s", n)

ahol s a sokszorozandó karakterlánc,

n egész szám, amely a többszörözések száma.

Hatása: az s karakterláncot n-szer írja egymásután.

Példa: Tekintsük az alábbi programsort

```
10 PRINT DUP ("XYZ3-SZOR", 3)
```

A végrehajtás eredménye a következő kiírás:

XYZ3-SZORXYZ3-SZORXYZ3-SZOR

Végrehajtása: a h tizenhatos számrendszerbeli számot tízes számrendszerbeli számmá alakítja át.

Példák:

a) `PRINT $00AB, $0C0A`

Képernyőn a következő számok jelennek meg a végrehajtás után:

171 3082

b) `X = $0CCC`

Az utasítás végrehajtása után az X változó értéke:

3276

2.5. Fényceruzával, paddle-vel és a botkormánnyal kapcsolatos függvények

2.5.1. Fényceruza koordinátái

A fényceruza x koordinátáját adja meg finom grafikában a `PENX` függvény.

Alakja:

`PENX`

Értéke: a fényceruza x koordinátája (0 és 320 között).

Az y koordináta értékét veszi fel a `PENY` függvény.

Alakja:

`PENY`

Értéke: a fényceruza y koordinátája (0 és 199 közötti szám).

2.5.2. Paddle potenciométerének állása

A paddle ellenállását adja meg a POT függvény.

Alakja:

$$POT(i)$$

ahol i a berendezés száma (0 v. 1).

Értéke: a paddle potenciométerének állása, 0 és 255 közötti érték.

2.5.3. Botkormány helyzete

A botkormány állásának és a tüzélgomb állapotának meghatározására alkalmas a JOY függvény.

Alakja:

$$JOY$$

Értékei:

- 128 tüzélgomb meg van nyomva
- 0 botkormány alapállásban
- 1 botkormány fel
- 5 botkormány le
- 3 botkormány jobbra
- 7 botkormány balra
- 8 botkormány fel balra
- 2 botkormány fel jobbra
- 6 botkormány le balra
- 4 botkormány le jobbra

3. Képernyő műveletek

3.1. Képernyőn szín villogtatás

A képernyőn egy szín villogtatására szolgál a **FLASH** utasítás.

Alakja:

FLASH sz, s

ahol sz szinkód,
s sebesség.

Hatása: a képernyő aktuális színét a megadott sz szinkóddal folyamatosan felcseréli a megadott sebességgel.

A sebesség 1 és 255 között változtatható.

Egy egységhez 1/16-od másodperc tartozik.

Megjegyzendő, hogy a **FLESH** utasítás a finom grafika esetén nem alkalmazható.

A **FLASH** utasítást, azaz a képernyő villogtatását az **OFF** utasítással hatástalaníthatjuk. Az utasításnak paramétere nincs, az aktuális **FLASH** utasítást hatástalanítja.

3.2. Keretszín villogtatása

A képernyő keretének színeit villogtatja a **BFLASH** utasítás.

Alakja:

BFLASH s, sz1, sz2

ahol s a sebesség,
sz1, sz2 szinkódok.

Hatása: az sz1, sz2 szinkóddal megadott színek felváltva villognak a kereten az s sebesség szerint.

Az sz1 és sz2 kódok értéke 0 és 15 közötti egész számok lehetnek. Az s sebesség értéke 0 és 255 közötti egész szám. Egy egység 1/16-od másodperc. A BFLASH hatása kikapcsolható a

BFLASH 0

utasítással.

Példa: Futtassuk le az alábbi programot:

```
10 BFLASH 8,2,3
20 FOR I=1 TO 2000:NEXT I
30 BFLASH 0
```

Hatására a képernyő kerete piros/zöld színekben villog, körülbelül három másodpercig.

3.3. Terület feltöltése karakterekkel

A képernyő egy részének megadott karakterrel való kitöltésére szolgál az FCHR utasítás.

Alakja:

FCHR a,b,c,d,e

- ahol
- a 0,24 közötti érték, sorszám,
 - b 0,39 közötti érték, oszlopszám,
 - c a terület szélessége karakterekben,
 - d a terület mélysége karakterekben,
 - e a területen megjelenítendő karakter POKE kódja.

Hatása: az a. sor b. oszlopában c karakter szélességű és d karakter mélységű területen kirajzolja az e POKE kóddal megadott karaktert.

Példa: A képernyő jobb felső sarkában egy 20x20-as négyzetet töl-
tünk fel Q betűkkel.

Megvalósítás:

```
10 PRINT "SHIFT/CLR/HOME" :POKE 53281,5
20 FCHR 0, 0, 19, 19, 17
```

3.4. Egy terület színének megválasztása

Egy területen megjelenő jelek színének megváltoztatására szolgál az FCOL utasítás.

Alakja:

FCOL a,b,c,d,e

ahol a 0, 24 közötti érték, a terület kezdetének sorszáma,
b 0, 39 közötti érték, a terület kezdetének oszlopszáma,
c a terület szélessége karakterekben,
d a terület mélysége karakterekben,
e 0 és 15 közötti érték, a területen megjelenítendő karakterek színe.

Hatása: a területen megjelenő karakterek színe a megadott c színű lesz.

Példa: A jobb felső 20x20 karakternyi területen megjelenő karakterek színe legyen piros.

Megoldás:

```
FCOL 0,0,19,19,2
```

3.5. Terület feltöltése színes karakterekkel

Egy megadott terület megadott színű karakterekkel való feltöltésére szolgál a `FILL` utasítás.

Alakja:

`FILL a,b,c,d,e,f`

ahol az első 4 paraméter (`a,b,c,d`) a terület megadására szolgál (lásd `FCHR`, `FCOL` utasítások),

`e` a területen megjelenítendő karakter `POKE` kódja,

`f` a területen megjelenítendő karakter szinkódja (0 és 15 közötti érték).

Példa: Legyen a képernyő bal felső 20x20-as négyzete sárga. A betűkkel feltöltve.

Megoldás:

`FILL 0,0,19,19,1,7`

3.6. Egy terület átmozgatása

Egy már a képernyőn levő terület máshova másolását végzi el a `MOVE` utasítás.

Alakja:

`MOVE a,b,c,d,e,f`

ahol az első négy paraméter (`a,b,c,d`) a másolandó terület megadására szolgál (lásd `FCHR`, `FCOL`),

`e` az új terület sorindexe,

`f` az új terület oszlopindexe.

Hatása: Az a. sor b. oszlopában kezdődő c,d karakternyi területet átmásolja a képernyő e. sorának f. oszlopától kezdődően.

Példa: Mozgassuk át a képernyő bal felső sarkában levő 10x10-es négyzetes területet a jobb alsó sarokba.

Megoldás:

```
10 PRINT "SHIFT/CLR/HOME"
20 FOR I=1 TO 10
30 PRINT"1234567890"
40 NEXT I
50 MOVE 1,1,10,10,15,30
```

3.7. Karakter inverz képének kiírása

Egy képernyőterületen a karakterek inverz képének megjelenítésére szolgál az INV utasítás.

Alakja:

INV a,b,c,d

ahol a négy paraméter a terület megadását szolgálja, amelyre az INV utasítás vonatkozik (lásd FCHR, FCOL).

Hatása: az a. sor b. oszlopától kezdődő c karakter szélességű és d karakter mélységű területen a karakterek inverz képe jelenik meg.

Példa: A képernyő bal felső sarkában egy 20x20-as négyzeten a karakterek inverz képe jelenjen meg.

Megoldás:

```
INV 0,0,19,19
```

A MOVE-nél elmondott példában a jobb alsó négyzet karaktereinek inverz képét a következő utasítás jeleníti meg:

```
40 INV 15,20,10,10
```

3.8. Képernyő egy területének rollozása

A képernyő kitüntetett sorainak illetve oszlopainak folyamatos "tekerését" végzi a következő utasítás. A mozgatus történhet jobbra, balra, fel, le.

Alakja:

- a) LEFT i a,b,c,d
- b) RIGHTi a,b,c,d
- c) UPi a,b,c,d
- d) DOWNi a,b,c,d

ahol az első szó az utasítás irányát jelölő kulcsszó,

i a tekerés tipusa, értékei W vagy B,

a,b az oszlop és a sor sorszámai, ahonnan a mozgatus kezdődik,

c,d a mozgatusandó terület utolsó oszlopának illetve sorának sorszámai.

Az i=W esetben a sorok (oszlopok) mozgatusa úgy történik, hogy a karakterek nem tűnnek el,

i=B esetben a sorokat nem "viszi" magával, azaz a mozgatusott terület helyén üres hely lesz.

Az oszlopok és sorok számozása itt is nullával kezdődik.

Hatása:

a) Jobbról-balra "rolloz" az a. oszlop b. sorától kezdve a c. oszlop b. soráig bezárólag. A sorok illetve oszlopok tartalma az i paramétertől függ.

b) Hatása ugyanaz, de a mozgatus balról jobbra történik.

- c) Hatása ugyanaz, de a mozgás alulról felfelé történik.
- d) Hatása ugyanaz, de a mozgás felülről lefelé történik.

Megjegyzés: az utasításokat nem használhatjuk finom grafikában!

Példa: a képernyő bal felső sarkán levő számok különböző irányu rollozását mutatjuk be az alábbi programmal.

```
10 PRINT "SHIFT/CLR/HOME"
10 FOR I=1 TO 10
20 PRINT I, I*, I*
30 NEXT I
40 FOR I=1 TO 100
50 LEFTW 0,0,10,10
60 NEXT I
70 FOR I=1 TO 100
80 RIGHTW 0,0,10,10
90 NEXT I
100 FOR I=1 TO 100
110 UPW 0,0,10,10
120 NEXT I
130 FOR I=1 TO 100
140 DOWNW 0,0,10,10
150 NEXT I
```

3.9. Képernyőtartalom kiírása

3.9.1. Képernyőtartalom kiírása háttértárolóra

Egy képernyő tartalmának háttértárolón való elhelyezésére szolgál az SCRSV utasítás.

Alakja:

- a) SCRSV 2,8,2,"n,S,W"
- b) SCRSV 1,1,1,"n,S,W"

ahol n a megőrzendő képernyő adatállomány neve a külső tárolón.

Hatás: az a) esetben az aktuális képernyőtartalmat mágneslemezre,
a b) esetben mágnesszalagra teszi ki n névvel.

A továbbiakban ezzel a névvel lehet hivatkozni és újra megjeleníteni az adatokat (például az SCRLD utasítással).

Megjegyzendő, hogy az utasítás nem használható finom-grafikus mód esetén.

3.9.2. Képernyőtartalom beolvasása háttértárolóról

A külső SCRSV utasítással kiírt képernyőtartalom visszahozását végzi a SCRLD utasítás.

Alakja:

- a) SCRLD 2,8,2,"n"
- b) SCRLD 1,1,1,"n"

ahol az a) és b) alakban írt számok megegyeznek az SCRSV utasításnál elmondottakkal,

n a tárolt képernyőtartalom azonosítója.

Hatása: az n néven SCRSV utasítással kiírt képernyőtartalmat a) lemezzről, b) szalagról visszahozza és megjeleníti.

Példa: az aktuális képernyőtartalmat írassuk ki lemezre ALMA néven, majd olvassuk vissza és jelentessük meg a képernyőn.

A feladatot az alábbi program végzi el:

```
10 SCRSV 2,8,2,"ALMA,S,W"  
20 PRINT "SHIFT/CLR/MOME"  
30 SCRLD 2,8,2,"ALMA"
```

3.9.3. Képernyőtartalom kiírása nyomtatón

A képernyő tartalmát kiirathatjuk sornyomtatóra a COPY utasítással.

Alakja:

COPY

Hatása: az aktuális képernyőtartalmat a gép kiírja a nyomtatóra.

Megjegyzendők:

- az utasítást csak grafikus Üzem módban használhatjuk,
- a CIRCLE utasítással kirajzolt kör, ellipszis a kiírásnál torzulhat, ezért figyelembe kell venni az ott elmondottakat.

Nem grafikus képernyőtartalom nyomtatón való kiírását végzi a HRDCOPY utasítás.

Alakja:

HRDCOPY

3.10. Kurzor helyének meghatározása

A kurzor képernyőn levő helyének értékét adja meg a LIN függvény.

Alakja:

LIN

Értéke: numerikus, annak a sornak a száma, amelyikben a kurzor jelenleg van.

Megjegyzendő, hogy a függvény értékét programból is lekérdezhetjük, és direkt módban is használható.

4. Programírást segítő parancsok

4.1. Automatikus sorszámozás

Az utasítások számának automatikus generálását, kiírását és növelését végzi el az `AUTO` parancs.

Alakja:

`AUTO s,n`

ahol `s` a kezdeti sorszám,
`n` a növekmény.

Tekintsük a következő programot:

```
1000 INPUT A,B
1100 X=A+B
1200 PRINT X
```

A fenti programot a következő lépésekben készíthetjük el az automatikus számozás felhasználásával:

Begépelés	Képernyő tartalma
<code>AUTO 1000,100</code>	1000
<code>INPUT A,B</code>	1000 INPUT A,B
<code>RETURN</code>	1000 INPUT A,B
	1100
<code>X=A+B</code>
	1100 X=A+B
<code>RETURN</code>
	1200
<code>PRINT X</code>
	1200 PRINT X
<code>RETURN</code>
	1300
<code>RETURN</code>	

4.2. Program átszámozása

A program átszámozását a RENUMBER parancs végzi.

Alakja:

RENUMBER s,n

ahol s az új kezdőszám,

n a növekmény.

Hatására: a gép átszámozza a programot úgy, hogy az első sor az s számot kapja, minden további számozott sor az előzőtől n-nel lesz nagyobb.

Megjegyezzük, hogy a programban szereplő GOSUB, GO TO utasítások utáni számokat a gép automatikusan nem cseréli. Az átszámozás miatt bekövetkező esetleges programhibákat a SIMONS BASIC-ban levő PROC utasítás használatával küszöbölhetjük ki.

Tekintsük a következő programrészletet

```
10 INPUT X
20 Y=X+X
30 GO TO 100
```

Számozzuk át a fenti programrészletet úgy, hogy az első utasítás száma 1000 legyen, a továbbiak pedig 100-zal növekedjenek.

Ezt a következő paranccsal végezzük el

RENUMBER 1000,100

A LIST parancs végrehajtása után a következő programot kapjuk:

```
1000 INPUT X
1100 Y=X+X
1200 GO TO 100
```

4.3. Programok összemásolása

Programok összemásolására szolgál a MERGE parancs. A gép egy a gépben lévő programhoz tud ezzel hozzámásolni egy valamilyen tárolón (lemez, szalag) elhelyezett programot.

Alakja:

MERGE "n", d

ahol n a mágneses tárolón elhelyezett program neve,

d az egységszám (1 esetén kazettás magnetofon, 8, ..., 15 esetén lemez).

Hatása: a gépben aktuálisan helyetfoglaló programhoz hozzámásolja az n nevű programot, amit korábban a d számú egységre írtunk ki.

Vigyázzunk az összemásolandó programok sorszámaira, ugyanis a másolandó programot egyszerűen a gépben lévő program mögé helyezi el.

4.4. Programlistázás vezérlése

4.4.1. Listázás oldalanként

Egy program listázása a LIST paranccsal történik úgy, hogy a gép folyamatosan írja ki a programokat. A program megadott soronkénti egységekben történő listázását végzi a PAGE parancs.

Alakja:

PAGE n

ahol n az egyszerre képernyőre irandó sorok száma.

Hatása:

- a programot az első sorától kezdve n soronként kiírja a LIST parancs hatására,
- a következő n sor kiírását a RETURN billentyű lenyomásával végezhetjük el.

A parancsot a PAGE 0 parancs kiadásával hatástalanítjuk.

Példa: Irassuk ki a gépben levő programot 10 soronként.

Ezt az alábbi parancsokkal érhetjük el:

```
PAGE 10/RETURN  
LIST/RETURN
```

A fenti két parancs hatására a gép kiírja a program első 10 sorát, a további sorokat a RETURN billentyű megnyomásával listázzhatjuk ki tízes egységekben.

4.4.2. Listázás lassítása

A listázás lassítását a CTRL billentyű lenyomva tartásával érhetjük el. A jelen verzió egy további lassítási lehetőséget nyújt a DELAY parancssal.

Alakja:

```
DELAY  $n$ 
```

ahol n 1 és 255 közötti numerikus érték.

Hatása:

- a SHIFT/LOCK billentyűt lenyomva a DELAY parancs hatására a program listázása lelassul,
- leggyorsabb a listázás $n=1$ esetén,
- $n=255$ esetén a listázás karakterenként történik (alapsza-

A billentyűt elengedve a listázás normál módban történik.

A `COMMODORE` billentyűt lenyomva a listázás megszakad, elengedve tovább folytatódik.

4.5. SIMON'S utasítások kiemelése

A `SIMON'S BASIC` egy sor "uj" utasítást tartalmaz a `BASIC`-hez képest. Az ezen utasítások megkülönböztetése lehetséges az eredeti utasításoktól az `OPTION` paranccsal.

Alakja:

`OPTION n`

ahol `n` 0 és 255 közötti érték lehet.

Hatása: `n=10` esetén a `SIMON'S BASIC` utasítások fordított (`REVERSE`) módban kerülnek kiíratásra listázásnál

`n` más megengedett értéke esetén a fordított mód kiíratás megszűnik.

4.6. Karakterlánc megkeresése

Egy `BASIC` programban előforduló tetszőszerinti karakterlánc helyének (helyeinek) megkeresésére szolgál a `FIND` parancs. Sok felhasználási területe közül csak egyet említünk meg: átsorszámozás esetén segítséget nyújt a `GO TO`, `GOSUB` utasítások megkeresésére és átsorszámozására!

Alakja:

`FIND c`

ahol `c` a keresendő karakterlánc.

A c értékei a következők lehetnek:

- a) karakter konstans idézőjelbe téve
- b) numerikus számjegy (számjegyek)
- c) BASIC alapszó
- d) változónév
- e) egyéb jel/ek/.

Hatása a fenti esetekben: kiírja azon sorok sorszámát, ahol

- a) a megadott karakterkonstans előfordul,
- b) az adott számjeggyel (számjegyekkel) kezdődő konstans előfordul,
- c) a megadott BASIC kulcsszó előfordul,
- d) a megadott betűvel (vagy jelekkel) kezdődő változó előfordul,
- e) egyéb jel/ek/ található/k/.

Példa: tekintsük a következő programrészletet:

```
10 INPUT A1:X=15
20 INPUT A$
30 PRINT A1, A$
40 A=A1*A1*1.5
50 PRINT "EREDMENY" , A1
60 GO TO 100
```

A FIND parancs végrehajtása

Parancs	Lista
FIND "EREDMENY"	50
FIND1	10,30,40,60
FIND PRINT	30,50

FINDA	10,20,30,40,50
FINDA1	10,30,40,50
FIND=	10,40
FIND=1	10
FINDTO	
FIND,	30,50

4.7. Programfutás nyomonkövetése

Egy program futása során szükséges lehet az utasítások végrehajtási sorrendjét ellenőrizni. Ezt végzi a **TRACE** parancs.

Alakja:

TRACE n

ahol **n** az utasítás paramétere,

n=0 esetben a **TRACE** befejezi a működését.

n=10, esetben parancs aktiv.

Hatása: A képernyő jobb felső sarkában egy területet lefoglal, ahova az éppen végrehajtott utasítás sorszámát írja be. Ha itt kiírási terület van, akkor ezt felülírja.

Ha a program befejezésekor néhány befejező utasítás sorszámát ki akarjuk iratni, akkor a **RETRACE** utasítást használjuk. A program befejezésekor ugyanis a képernyőről az utasítássorszámok eltűnhetnek a **TRACE** utasítás alkalmazása esetén.

Alakja:

RETRACE

4.8. Változók tartalmának kiírása

A gépben levő program változói tartalmának a kiírására is lehetőség van a `DUMP` paranccsal.

Alakja:

`DUMP`

Hatása: A gép képernyőjén megjelennek a program változóinak nevei és értékei egyenlőségjellel összekötve. Amennyiben a változók száma 25-nél több, úgy a `CTRL` billentyűt lenyomva tartva kaphatjuk meg részenként a teljes listát.

Példa: Futtassuk le a következő programot:

```
10  A=25
20  B=A*2
30  C$="ALMA A FA ALATT"
  '
```

Adjuk ki a `DUMP` parancsot.

A képernyőre a következők kerülnek kiírásra

```
A=25
B=50
C$="ALMA A FA ALATT"
```


5. Saját grafikus alakzatok készítése

5.1. Sprite manipuláció

Sprite, un. szellem /S/ létrehozása egyszerű utasításokkal lehetséges

5.1.1. Helyfoglalás a memóriában

Az /S/ helyfoglalás - a MOB - részére a DESIGN utasítással lehetséges.

Alakja:

- a) DESIGN a,b
- b) DESIGN a,b+c

ahol

<u>a</u>	a típus jelölés, értékei 0 vagy 1,
<u>b</u>	S részére lefoglalt tárolóhely első byte-jének címe,
<u>c</u>	finom grafika esetén használatos konstans, értéke 49152 (v. 8C000 hexadecimálisként).

Az a paraméter hatása

a=0 esetén finom grafika

a=1 esetén színes (multikolor) grafika

A b értékei:

- 64 többszörösének kell lennie,
- decimálisként vagy hexadecimálisként kell megadni, ez utóbbi esetben 8 jellel kezdődik.

A kezdőcímek a következők lehetnek:

blokkszám	memóriahely
13-15	832-1023
32-63	2048-4095
128-255	8192-16383

Ha a MEM utasítást is használjuk, akkor csak a 192-255 blokkok állnak rendelkezésre.

5.1.2. Szellem megrajzolása

Szellem megrajzolásához ad segítséget a @ eljárás.

Alakja:

- a) @
- b) @

A fenti pontok helyébe üthetjük be a szellem alakját kirajzoló karaktereket. Az a) formát a multikolor, a b)-t a finom grafika esetén alkalmazzuk. Mindkét esetben az egy alakzathoz tartozó sorok száma 21.

A beütendő karakterek értéke:

a) esetben, azaz multikolor módban

szinkód	értelmezése
B	a CMOB utasítás első színe
C	a MOB SET utasításban megadott szín
D	a CMOB utasításban megadott második szín

b) esetben

szinkód

értelmezése

B

a MOB SET utasításban
megadott szín

Természetesen mindkét esetben a kipontozva maradt terület a háttérszint veszi fel.

5.1.3. Szines sprite szineinek megadása

A multikolor módban a színek megadását a CMOB utasítás végzi .

Alakja:

CMOB s1, s2

ahol s1, s2 a megadott színek kódja (0-15 közötti érték).

5.1.4. Szellem felépítése

A "szellem" paramétereit adhatjuk meg a MOB SET utasítással .

Alakja:

MOB SET a,b,c,d,e

ahol a a megadandó szellem száma, a kisebb számú nagyobb prioritású, azaz eltakarja mozgatsnál az alacsonyabb prioritásút,

b a blokkszám, ahol a szellem adatai vannak,

c a szín megadása, amely színes szellem esetén a C, más esetben a B szinkódot jelöli ki,

d prioritás, amely hatására

$d=0$ érték esetén a szellem magasabb prioritású a képernyő egyéb jeleivel szemben,

$d=1$ esetben a helyzet fordított,

e típus kijelölése

$e=1$ esetben színes,

$e=0$ esetben finom grafika.

5.1.5. Szellem megjelenítése és mozgatása

A szellem képernyőn történő megjelenítésére és mozgatására szolgál az MMOB utasítás.

Alakja:

MMOB $a, x1, y1, x2, y2, b, c$

ahol a a szellem száma,

$x1, y1$ a megjelenítés helyének koordinátái,

$x2, y2$ azon pont koordinátái, ahova a mozgatás történik,

b a szellem méretének megadására szolgáló paraméter;

c mozgatási sebesség (1-255 közötti érték).

Hatása: a szellem képe az $x1, y1$ pontból az $x2, y2$ pontba mozog b -ben megadott méretben c sebességgel.

Megjegyzések:

1. Az $x1, y1, x2, y2$ koordináták a szellem bal felső "sarkának" koordinátái.

2. A b méret az alábbi értékeket veheti fel:

érték	hatás
0	normál méret
1	x irányban kétszeres nagyítás
2	y irányban kétszeres nagyítás
3	mindkét irányban kétszeres méretnövekedés.

3. A mozgási sebesség ellentétes a máshol megadott sebességgel:

1	a leggyorsabb sebesség
255	a leglassabb sebesség.

A már a képernyőn levő szellem mozgását végzi az RLOCMOB utasítás.

Alakja:

RLOCMOB a,x,y,b,c

ahol a megegyezik az MMOB utasításnál elmondottakkal,
x,y azon hely koordinátái, ahova a szellemet mozgatjuk,
b,c paraméterek megegyeznek a MMOB utasításnál elmondottakkal.

Hatása: A szellem a képernyőn lévő aktuális helyéről az x,y pontokba megy a b mérettel és a c sebességgel.

5.1.6. Alakzatok összeköttetésének érzékelése

Az alakzatok összeköttetésének lekérdezését készíti elő a DETECT utasítás.

Alakja:

DETECT n

ahol n az összeütközés típusának megadására szolgáló paraméter.

Az n értékei:

$n=0$ esetben két szellem közötti összeütközés lekérdezésének előkészítését írjuk elő,

$n=1$ esetben egy szellem és a képernyőn levő egyéb karakterek összeütközése lekérdezésének előkészítését írjuk elő.

Megjegyzések:

1. A DETECT utasítást a CHECK utasítás előtt alkalmazzuk.

2. A DETECT utasítást kétszer kell kiadni:

- az első végrehajtása esetén az összeütközést jelző regiszter tartalma törlődik,
- a második végrehajtásánál a fenti regiszter az összeütközés regisztrálására készen áll.

A CHECK kérdezi le az összeütközés regiszter tartalmát.

Alakja:

a) IF CHECK (m_1, m_2) = 0 THEN u

b) IF CHECK (0) = 0 THEN u

ahol m_1, m_2 szellemek számai,

u végrehajtható utasítás.

Hatása:

- a) Lekérdezi, hogy az m1 és m2 szellem összeütközése bekövetkezett-e.
- b) Lekérdezi, hogy a szellem valamely képernyőn levő karakterrel összeütközött-e.

Ha összeütközés történt bármely (a) és (b) esetben, akkor az u utasítás hajtódik végre, egyébként az IF utasítást követő utasítás.

5.1.7. Szellem kitörlése

A képernyőn levő szellem kitörlését végzi a MOB OFF utasítás.

Alakja:

MOB OFF a

ahol a a képernyőről kitörlendő szellem száma.

5.2. Saját karakterek készítése

A saját karakterek készítéséhez először át kell másolni a karakter ROM-ot RAM-ba (MEM utasítás). Ezután meghatározzuk a DESIGN utasítással azt a karaktert, amit ki akarunk cserélni. Ezután szerkesztjük meg a @ utasítással a kívánt karaktert. Az alábbiakban ezeket az utasításokat mutatjuk be.

5.2.1. Memória átmásolása

A memóriaterület ROM memóriájában elhelyezett karakterkészlet RAM memóriaterületre történő átmásolását végzi a MEM utasítás.

Alakja:

Hatása: A C64 karakterkészlete RAM területre másolódik.

Megjegyzések:

1. A RAM területen ezután tetszésszerűen saját karaktert tehetünk a régié helyébe.
2. Áttolja a képernyő RAM-ot a \$C000 címűtől kezdve (a kernél mögé).
3. Bekorlátozza a MOB területet a 192-től a 255-ös blokkokra.

5.2.2. Kicserélendő karakter megadása

A kicserélendő karaktert adjuk meg a DESIGN utasítás segítségével.

Alakja:

DESIGN 2, \$E000+C*8

ahol C a kicserélendő karakter poke kódja.

Hatása: a poke kóddal megadott karaktert kicserél a @ -ban megadott "formátumra", amit alább ismertetünk.

A @ segítségével szerkeszthetjük meg az új karaktert.

Alakja:

@

Megjegyzések:

1. Összesen 8 sort kell megadni egy karakter megszerkesztéséhez.
2. A helyére B betűt kell írni oda, ahol a karaktert meg akarjuk szerkeszteni.

Példa: egy karakter megszerkesztése:

50	@	.	B	.	B	B	.	B	.
60	@	.	B	.	B	B	.	B	.
70	@	.	B	B	.
80	@	.	B	B	.
90	@	.	B	B	.
100	@	.	B	B	.
110	@	.	.	B	.	.	B	.	.
120	@	.	.	.	B	B	.	.	.

A fenti karaktert a @ karakter helyett alkalmazzuk. Ehhez a következő utasításokat kell végrehajtani:

10	MEM
20	DESIGN 2,8E000+0*8

5.3. Példa MOB szerkesztésére, mozgatására, színezésére

Készítsük el és futtassuk le az alábbi programot:

```
10  DESIGN 1,11*64
20  @ .....
30  @ .....
40  @ .....
50  @ .....
60  @ BBBBBBBBBBBB.
70  @ .....B.....
80  @ .....B.....
90  @ ....BBB.....
100 @ ..DDDC.....
110 @ ..D..C C....
120 @ .D...C.C....
130 @ .D...C.C....
140 @ D....C.C....
150 @ D....C.C...B
160 @ BBBBBBBB..BB
170 @ BBBBBBBBBBBB
180 @ BBBBBBBBBB...
190 @ ..B..B.....
200 @ .B....B.....
210 @ .....
220 @ .....
230 MOB SET 1,11,5,1,1:PRINT:SHIFT/CLR/HOME"
240 FOR T=1 TO 100
250 RLOCMOB 0,RND(1)*300+20,RND(1)*180+20,RND(1)*4,
    RND(1)*80+1
260 CMOB RND(1)*15,RND(1)*15:NEXT
```

6. Zenei utasítások

A C64 BASIC-ben az egyes zenei paramétereket a POKE utasítás felhasználásával vezérelhetjük. A SIMON'S BASIC lehetővé teszi a paraméterek utasításban való megadását is.

6.1. Hangerő szabályozása

A hangerősség megadását teszi lehetővé a VOL utasítás.

Alakja:

VOL n

ahol n a hangerősség megadására szolgáló paraméter.

Az n értéke 0-15 között változhat a következőképpen:

n=0 esetén "lekapcsoltuk" a hangot,

n=1 minimális hangerő,

.

n=15 maximális hangerő.

6.2. Hullámforma és egyéb paraméterek

Különböző paraméterek megadását érhetjük el a WAVE utasítással.

Alakja:

WAVE h, b

ahol h a hang megadására szolgáló paraméter,

b egy 8 jegyű kettes számrendszerbeli szám.

A	h	értékei:	1	1. hangra vonatkozik
			2	2. hangra vonatkozik
			3	3. hangra vonatkozik

A b egyes helyértékei egy-egy hang jellemző megadására szolgálnak. Ha az aktuális helyen 1 van, akkor a megfelelő effektus aktív, egyébként nem.

Az egyes helyértékek a következőket jelentik (balról jobbra):

1. hely	zajgenerátor
2. hely	négyszög hullámforma
3. hely	fűrészfog hullámforma
4. hely	háromszög hullámforma
5. hely	nem használatos
6. hely	háromszög hullámforma esetén ring moduláció
7. hely	szinkronizáció
8. hely "kapu" bit,	a SIMON BASIC automatikusan beállítja

6.3. Burkoló görbe megadása

A burkoló görbe paramétereinek megadását végzi az 1-3 hang esetén az ENVELOPE utasítás.

Alakja:

ENVELOPE h, A, D, S, R

- ahol
- h a hang paraméter (1, 2 vagy 3)
 - A felfutás maximuma (Attach)
 - D lefutás az egyenletes erősségre (Decay)
 - S egyenletes erősség ideje (Sustain)
 - R lefutás nullára (Release).

6.4. Lejátszandó zene megszerkesztése

A zene tempójának és hangjainak megadására szolgál a MUSIC utasítás.

Alakja:

MUSIC n,"c"

ahol n az un. időtartam paraméter, a tempó megadására szolgál (1 és 255 közötti érték),
c lejátszandó zene megadására szolgáló karakterlánc.

A c tartalmazza a lejátszandó zenei hangokat. Első karaktere azt mondja meg, hogy melyik hangot szólaltatja meg (azaz az egyest, kettest vagy hármast). Ezután kell megadni a lejátszandó zenét. Minden egyes hangnál meg kell adni a

- hang hosszát egy vezérlő karakterrel,
- a hang nevét (C-H),
- az oktávot (0-7).

A vezérlő karakter értékeit és jelentését az alábbiakban foglaljuk össze:

vezérlő karakter	jelentése
SHIFT/CLR/HOME+1v.2v.3	a hang megadására szolgáló vezérlőjel
F1	következő hang 1/16 ideig szól
F3	következő hang 1/8 ideig szól
F5	következő hang 1/4 ideig szól
F7	következő hang 1/2 ideig szól
F2	következő hang 1/1 ideig szól
F4	következő hang 2/1 ideig szól
F6	következő hang 4/1 ideig szól
F8	következő hang 8/1 ideig szól

A hang és az oktáv megadásához a nevét az oktáv számát kell megadni.

Például: C5, D5, F2, E3

6.5. Zene lejátszása

A MUSIC utasításban megadott zene lejátszására szolgál a PLAY utasítás.

Alakja:

PLAY n

ahol n értéke 0, 1 vagy 2 lehet és a zene lejátszásának "módját" adja meg.

Az n értékeire és jelentésére tekintsük az alábbi táblázatot:

n értéke	jelentése
0	zene leállítása
1	megadott zene lejátszása a program futásának felfüggesztésével
2	megadott zene lejátszása a program párhuzamos futtatásával

A zenei lehetőségek bemutatására tekintsük az alábbi egyszerű miniprogramot:

```
150 VOL 10
170 WAVE 1,00100000
190 ENVELOPE 1,0,15,10,0
210 A$="C12C5E5F5"
220 A2$="G5C5E5F5G5C5E5F5G5"
230 A2$=A2$+"E5C5E5D5E5E5D5C5"
240 A2$=A2$+"C5E5G5G5F5F5E5F5"
250 A2$=A2$+"G5E5C5D5C5C5C5E5"
260 A2$=A2$+"F5"
270 A3$="C5C5"
280 MUSIC 8,A$+A2$+A2$+A3$
300 PLAY 1
320 VOL 0
330 END
```

READY.

7. Egyéb parancsok

7.1. Disk műveletek

A SIMON BASIC-ben a lemezműveletekhez szükséges parancsok köre is bővül. A disk műveletek egy része helyettesíthető a DISK parancscsal.

Alakja:

DISK "művelet"

ahol a művelet logikai állományok megnyitása, egységszám, második cím megadása, PRINT logikai állomány, program el-
tevése stb. lehet.

Példák:

1. Lemez header készítés TESZT néven

DISK "NO:TESZT,01"

2. Lemezen levő ALMA nevű program törlése

DISK "SO:ALMA"

7.2. Lemezállományok neveinek listázása

A lemezen levő állományok neveinek listázását végezhetjük a DIR parancscsal.

Alakja:

- a) DIR "g
- b) DIR "g:sk
- c) DIR "g:ksk

ahol $\$$ dollár jel,
k kérdőjel vagy kérdőjel sorozat
s karakterlánc.

Hatása:

- a) Kiírja a gépben levő lemez direktoriát (lásd LOAD "\$",8).
- b) Kiírja az s karakterláncsal kezdődő állományokat.
- c) Kiírja azon állományokat, amelyek nevében a kérdőjelek számát meghaladó karakterek megegyeznek s-sel.

Példa: Irassuk ki azon állományok neveit és helyfoglalását, amelyek nevének 3. és 4. karaktere PQ

Megoldás:

DIR "\$:??PQ*

A DIR paranccsal számos egyéb "feladat" is megoldható. A 4 karakter, illetve a 9 karakter hosszúságú programok nevét a következő utasításokkal írhatjuk ki:

DIR "\$:????

DIR "\$:??????????

7.3. NEW parancs hatástalanítása

Amennyiben kiadtuk a NEW parancsot, de valamilyen okból mégis szükségünk van az utoljára a gépben levő programra, akkor az OLD paranccsal hozzáférhetünk.

Alakja:

OLD

azaz paramétere a NEW parancshoz hasonlóan nincs. Egy alkalmazást megemlítünk: amennyiben a COLD parancsot használjuk, ez a gépben lévő aktuális programot is "kitörli". Az OLD parancssal azonban ez is aktivizálható.

7.4. Funkció billentyűk használata

Funkció billentyűk parancshoz rendelése.

Egy parancs funkció billentyűhöz való rendelését végzi a KEY utasítás.

Alakja:

KEY s, "c"

ahol s a funkció billentyű száma ($1 \leq s \leq 16$)
c utasításnév.

Az utasításnév legfeljebb 16 karakter hosszú lehet. A KEY parancsot a kulcsszó legépelésével és a RETURN billentyű lenyomásával lehet végrehajtani. A továbbiakban a megfelelő funkció billentyűt lenyomva a hozzátartozó c szöveg jelenik meg a képernyőn.

Példa:

KEY 1, "LOAD"

Az utasítás végrehajtása után az F1 billentyűt lenyomva a LOAD kiírás jelenik meg a képernyőn.

A kiírt parancs végrehajtása csak akkor történik meg, ha a RETURN billentyűt is lenyomjuk. Ezen billentyű "lenyomását" is hozzárendelhetjük a KEY utasítással valamelyik funkcióbillentyűhöz úgy, hogy

Példa:

KEY 1,"LOAD"+CHR\$(13)

A fenti parancs végrehajtása után az F1 billentyű megnyomása ekvivalens lesz a következő billentyűk megnyomásával:

L, O, A, D, RETURN

A képernyőre kiirathatjuk az egyes funkció billentyűkhöz rendelt utasítás - neveket is a DISPLAY paranccsal.

Alakja:

DISPLAY

A parancs végrehajtása a RETURN billentyű lenyomásával történik.

Tételezzük fel, hogy a fenti KEY parancsot már végrehajtottuk, és a DISPLAY parancsot begépeztük. A gép képernyőjén a következő szöveg jelenik meg:

DISPLAY:KEY 1,"LOAD"+CHR\$(13)

KEY2,""

KEY3,""

KEY4,""

KEY5,""

KEY6,""

KEY7,""

KEY8,""

KEY9,""

KEY10,""

KEY11,""

KEY12,""

KEY13,""

KEY14,""

KEY15,""

KEY16,""

Billentyűk használata:

F1, F3, F5, F7

billentyű megnyomása,

F2, F4, F6, F8

F1,F3,F5,F7 billentyű és SHIFT

F9, F11, F13, F15

**F1,F3,F5,F7 billentyű és a COMMODORE
billentyű**

F10, F12, F14, F16

**F1,F3,F5,F7 billentyű és SHIFT és
COMMODORE billentyű**

III.

SUPERGRAPHIK 64

Bevezetés

A rajzolási lehetőségek kényelmesebbé tételére készült a

SUPERGRAPHIK 64

software anyag. A gyártó a DATA BECKER cég.

Két változat létezik:

- a) mágnesszalagos
- b) mágneslemezes

A mágnesszalagos változat valamivel kevesebb lehetőséget tartalmaz, mint a mágneslemezes változat. A fő különbség az, hogy a b) változat zenei utasításokkal és az un. 80x50-es grafikával bővült.

Mivel hazánkban a szalagos változat terjedt el elsősorban, ezért ezt ismertetjük részletesen. A lemezes változat két eltérő utasításcsoportját (zene, 80x50-es grafika) részletesen ismertetjük, a többi utasítást azonban csak vázlatosan.

1. Szalagos változat

1.1. Rendszer betöltése

Betöltéséhez az alábbi műveletek szükségesek:

1. Helyezzük a SUPERGRAFIK64 kazettát a kazettás egységbe
2. Gépeljük be a

LOAD

parancsot.

A gép válasza:

PRESS PLAY ON TAPE

3. Néhány fordulat (~7) után a képernyőn megjelenik a

FOUND GRAFIK 64

Üzenet.

4. Majd a képernyő megint eltűnik és a program betöltése után ki-
írja a

CBM64 HI-RES GRAPHICS SOFTWARE REV.F

/C/ BY DATA BECKER GMBF

CREATED 19/01/1983

BY M. THIELKER

GRAPHICS READY

READY

Üzenetet.

1.2. Rajzolási mód megadása

Rajzolási mód megadása nélkül nem jelenik meg rajz a képernyőn.
Ezért a rajzolás megkezdése előtt ki kell adni a mód megadására
szolgáló GRAPHICS parancsot.

Alakja:

!GRAPHICS p

ahol **p** a rajzolósi paraméter.

A **p** a következő értékeket veheti fel:

p	jelentése
0	a grafikus képernyőről vissza- állunk normál módra
1	HR grafika kapcsolódik be, a további utasítások erre vonatkoz- nak
2	megegyezik a p=1 -gyel
3	a HR grafika bekapcsolódik, a következő utasítások azonban a normál képernyőre vonatkoznak
4	megegyezik a p=3 esettel
5	a HR grafikus képernyő kikapcsoló- dik, de az utasítások erre vonatkoz- nak, a grafikus kép akkor jelenik meg újra, ha kiadjuk a GRAPHICS 1 parancsot

A fenti táblázatban a HR (high resolution graphics) a grafikus mód - dot jelenti.

Az utasítás részletes értelmezését a következő példákban mutatjuk be.

1.3. Grafikus képernyő törlése

Az aktuális grafikus képernyő tartalmának kitörlésére szolgál a GCLEAR utasítás.

Alakja:

!GCLEAR

1.4. Geometriai alakzatok megjelenítése

1.4.1. Egy pont megjelenítése, törlése

Egy pont megjelenítésére szolgál a DOT utasítás.

Alakja:

!DOT x,y

ahol x,y a megjelenítendő pont koordinátái.

Egy pont törlésére szolgál a CDOT utasítás.

Alakja:

!CDOT x,y

ahol x,y a kitörlendő pont koordinátái.

Példa: Tekintsük a következő programot:

```
10  !GCLEAR:REM KEPERNYO TORLES
20  !GRAPHICS 1: REM MOD MEGADAS
30  FOR T=1 TO 100
40  !DOT 2*T,T : REM 1.EGYENES RAJZOLAS
50  !DOT 2*T, T+5 : REM 2.EGY.RAJZ.
60  !DOT 2*T, T+10 : REM 3.EGY.RAJZ.
70  !DOT 2*T, T+15 : REM 4.EGY.RAJZ.
80  !DOT 2*T, T+20 : REM 5.EGY.RAJZ.
90  NEXT T
```

A fenti program 5 párhuzamos egyenest rajzol a képernyőre.
A legutolsó vonal felső felét törli ki a következő programrészlet:

```
100  FOR T=51 TO 0 STEP-1
110  !CDOT 2*T, T+20 : REM TORLES
120  NEXT T
```

■

1.4.2. Egyenes szakasz kirajzolása és törlése

Egyenes kirajzolását érhetjük el a `LINE` utasítással.

Alakja:

`!LINE x1,y1 TO x2,y2`

ahol `x1,y1` a kirajzolandó szakasz kezdőpontjának koordinátái,

`x2,y2` a kirajzolandó szakasz végpontjának koordinátái.

Példa: Tekintsük a következő programot

```
10  !GCLEAR
20  !GRAPHICS 1
30  FOR J=0 TO 199 STEP 5
40  !LINE 0,J TO 319,J:REM EGYENES RAJZ.
50  NEXT J
```

A fenti program vízszintes vonalakkal rajzolja tele a képernyőt.

Ha függőleges vonalakkal is tele akarjuk "írni" a képernyőt, akkor a következő programot kell még lefuttatnunk:

```
60  FOR I=0 TO 319 STEP 5
70  !LINE I, 0 TO I,199
80  NEXT I
```

Az egyenes szakasz törlését végzi el a `!CLINE` utasítás.

Alakja:

`!CLINE x1,y1 TO x2,y2`

ahol `x1,y1` a törlendő egyenes kezdőpontjának koordinátái,
`x2,y2` a törlendő egyenes végpontjának koordinátái.

Példa:

A fenti programmal kirajzolt vízszintes és függőleges egyenesek minden másodikát törölhetjük ki, ha a következő programot futtatjuk le:

```
90  FOR J=0 TO 199 STEP 10
100  !CLINE 0,J TO 319,J
110  NEXT J
120  FOR I=0 TO 319 STEP 10
130  !CLINE I, 0 TO I,199
140  NEXT I
```

Szaggatott vonallal rajzolhatjuk ki a szakaszt a `DLINE` utasítással.

Alakja:

`!DLINE x1,y1 TO x2,y2`

ahol `x1,y1` a szakasz kezdőpontjának koordinátái,
`x2,y2` a szakasz végpontjának koordinátái.

A grafikus kurzor aktuális pontjától rajzol egy szakaszt a megadott pontig a `DRAW` utasítás.

Alakja:

`!DRAW x,y`

ahol `x,y` a megrajzolandó egyenes szakasz végpontja.

Megjegyezzük, hogy a grafikus kurzort rajzolás nélkül is tudjuk mozgatni. Erre szolgál a `MOVE` utasítás.

Alakja:

`!MOVE x,y`

ahol `x,y` az új pont koordinátái, ahova a grafikus kurzor mozgása történik (természetesen rajzolás nélkül!).

1.4.3. Téglalap kirajzolása, eltörlése

Egy téglalapot rajzolhatunk ki a `FILL` utasítás segítségével, amely egyben be is festi az adott rajzolási színnel a téglalapot.

Alakja:

`!FILL x1,y1 TO x2,y2`

ahol `x1,y1` a téglalap bal felső sarkának koordinátái,
`x2,y2` a téglalap jobb alsó sarkának koordinátái.

Megjegyzések:

1. A téglalap oldalai párhuzamosak a rajzolósi területtel (képernyő oldallapjai).
2. A kitöltési szín alapállapotban fehér.
3. Színes téglalapokat a LCOL és a BCOL utasítások előzetes használatával rajzolhatunk (lásd későbbi pontban).

Ha nem akarjuk a téglalapot befesteni, csak az oldalait kirajzolni, akkor a FRAME utasítást használjuk.

Alakja:

!FRAME x1,y1 TO x2,y2

ahol x1,y1 a kirajzolandó téglalap bal felső sarkának koordinátái,
x2,y2 az ezen téglalap jobb alsó sarkának koordinátái.

A FILL utasítással kirajzolt "befestett" téglalap eltüntetésére szolgál a CLEAR utasítás.

Alakja:

!CLEAR x1,y1 TO x2,y2

ahol x1,y1 a kitörölni kívánt téglalap bal felső sarkának koordinátái,
x2,y2 az ugyanezen téglalap jobb alsó sarkának koordinátái.

A FRAME utasítással kirajzolt téglalap kitörölését végzi a CFRAME utasítás.

Alakja:

!CFRAME x1,y1 TO x2,y2

ahol az utasítás paraméterei megegyeznek a CLEAR utasítás paramétereivel.

Példa:

A fenti utasításokat felhasználva rajzoljunk ki egy sakktáblát.

A feladatot a következő programmal oldjuk meg:

```
10  !GCLEAR
20  !GRAPHICS 1
30  !FRAME 0,20 TO 160,180
35  FOR I=1 TO 8 STEP 2
40  FOR T=0 TO 140 STEP 40
50  !FILL T,I*20 TO T+20,I*20+20
60  NEXT T
70  FOR T=20 TO 160 STEP 40
80  !FILL T,I*20+20 TO T+20,I*20+40
90  NEXT T
100 NEXT I
```

1.4.4. Szabályos görbevonalu alakzatok rajzolása

Kör, ellipszis rajzolására szolgál a CIRCLE utasítás. Ezzel kirajzolhatjuk az alakzat egy részét is.

Alakja:

!CIRCLE x1,y1,x2,y2 /s1,s2/

ahol x1,y1 a kirajzolandó alakzat középpontja,

x2 az X irányu sugár hossza,

y2 az y irányu sugár hossza,

s1 a rajzolás kezdőszöge fokokban,

s2 a rajzolás "vég"-szöge fokokban.

Példa:

A képernyő közepére rajzol ki 5 koncentrikus kört a 159, 99 koordinátákkal megadott középponttal az alábbi program:

```
10  !GCLEAR
20  !GRAPHICS 1
30  FOR T=0 TO 100 STEP 20
40  !CIRCLE 159, 99, T,T
50  NEXT T
```

A szögek használatára módosítsuk a programot az alábbi utasításokkal:

```
30  FOR T=0 TO 100 STEP 5
40  !CIRCLE 159, 99, T, T, 100-T, 260+T
```

A lefuttatás után egy csepp alakot hagy ki a koncentrikus körök felső részéből a T-vel módosított szögeknek megfelelően.

A CIRCLE utasítással kirajzolt körök (ellipszisek) törlését végzi a CCIRCLE utasítás.

Alakja:

```
!CCIRCLE x1,y1,x2,y2 /,s1,s2/
```

ahol a megadott paraméterek megegyeznek a CIRCLE utasításnál megadott paraméterekkel.

Példa:

Töröljük ki a fenti programmal megrajzolt körök minden másodikát.

Megoldás:

```
60  FOR T=0 TO 100 STEP 10
70  !CCIRCLE 159, 99, T, T, 100-T, 260+T
80  NEXT T
```

A `CIRCLE` utasítással kirajzolható alakzatokat szaggatott vonallal rajzolja ki a `DCIRCLE` utasítás.

Alakja:

```
!DCIRCLE x1,y1,x2,y2 /,s1,s2/
```

ahol az utasítás paraméterei megegyeznek a `CIRCLE` utasítás paramétereivel.

Példa:

Rajzoljuk ki szaggatott vonallal közös középpontú ellipsziseket.

Az alábbi program oldja meg a feladatot:

```
10  !GCLEAR
20  !GRAPHICS 1
30  FOR T=20 TO 100 STEP 20
40  !DCIRCLE 159, 99, T, T-20
50  NEXT T
```

A futtatás eredménye 5 ellipszis, amelynek a középpontja a 159, 99 pont, és a kirajzolás szaggatott vonallal történik.

1.5. Színezési utasítások

1.5.1. Háttérszín megadása

A képernyő háttérszínének megadására szolgál a `BCOL` utasítás.

Alakja:

```
!BCOL n
```

ahol `n` a képernyő hátterének színe.

Emlékeztetőül a szinkódok:

0	fekete	8	narancs
1	fehér	9	barna
2	piros	10	világos piros
3	türkiz	11	1.sz.szürke
4	lila	12	2.sz.szürke
5	zöld	13	világoszöld
6	kék	14	világcskék
7	sárga	15	3.sz.szürke

1.5.2. Rajzolósi szín megadása

Az alakzatok kirajzolásához megadhatjuk a rajzolósi színt az LCOL utasítással.

Alakja:

```
!LCOL n
```

ahol n a rajzolósi szín (0 - 15 közötti érték).

Hatása: az alakzatok egészen addig a megadott színnel lesznek kirajzolva, ameddig újabb LCOL utasítás nem következik.

Példa: Tekintsük az alábbi programot

```
10  !GCLEAR
20  !GRAPHICS 1
30  FOR T=0 TO 15
40  !BCOL T
50  !LCOL T
60  !FILL T*20,0 TO T*20+20,199
70  NEXT
```

Eredmény: a háttérszínek végigskálázzák az összes szint, miközben minden váltásnál egy újabb függőleges színes téglalap rajzolódik az előzőek mellé.

A téglalapok végül kitöltik az egész képernyőt, a színek sorrendje megegyezik a növekvő numerikus értékekhez tartozó színekkel.

Megjegyzés: a keretszint továbbra is a POKE 53280,i utasítással adhatjuk meg!

Ha a keretszint is meg akarjuk változtatni a háttérszín színére, akkor az 50-es utasítást cseréljük ki az alábbira:

50 ILCOL T : POKE 53280,T

1.6. Szellem grafikai utasítások

A szellem kirajzolásához szükséges alakzat megadását szolgálja a SPRITE utasítás.

Alakja:

!SPRITE n,v

ahol n a szellem száma (0-7 közötti érték),

v karakter típusu változó, amely a szellem adatait tartalmazza (63 byte hosszúságú).

A szellem méretének és színének megadását végzi az SMODE utasítás.

Alakja:

!SMODE n,s,p

ahol n a szellem sorszáma

s a megjelenítés nagysága

- $s = 0$ normál nagyság,
- $s = 1$ x irányban kétszeres nagyítás,
- $s = 2$ y irányban kétszeres nagyítás,
- $s = 3$ minden irányban kétszeres nagyítás,
- p szellem színének megadása (0-15 közötti egész szám).

Szellemgrafika mozgatóására szolgál az SMOVE utasítás.

Alakja:

!SMOVE n, x, y

ahol n a szellem sorszáma,

x, y a mozgató végpontjának koordinátái.

Példa: Az alábbi program a fenti utasítások használatát mutatja be.

Az alakzat megszerkesztése után nagyítjuk x majd y irányba, ezután mindkét irányba. Bemutatjuk a színek változtatásának módját és a mozgató lehetőségét is.

```

1   IBCOL1
10  ADAT$=""
20  FORT=0T062
30  READ ADAT
40  ADAT$=ADAT$+CHR$(ADAT)
50  NEXT
60  IGCLEAR
70  IGRAPHICS1
80  ISPRITE0,ADAT$
86  FORJ=0T03
90  FORT=0T0199
100 ISMODE 0,J,J+2
110 ISMOVE0,T,T

```

120	NEXT
130	NEXT
1000	DATA 255,Ø,255
1010	DATAØ,255,Ø
1020	DATA255,Ø,255
1030	DATAØ,255,Ø
1040	DATA255,Ø,255
1050	DATAØ,255,Ø
1060	DATA255,Ø,255
1070	DATAØ,255,Ø
1080	DATA255,Ø,255
1090	DATAØ,255,Ø
1100	DATA255,Ø,255
1110	DATAØ,255,Ø
1120	DATA255,Ø,255
1130	DATAØ,255,Ø
1140	DATA255,Ø,255
1150	DATAØ,255,Ø
1160	DATA255,Ø,255
1170	DATAØ,255,Ø
1180	DATA255,Ø,255
1190	DATAØ,255,Ø
1200	DATA255,Ø,255
1210	DATAØ,255,Ø

1.7. Egyéb utasítások

1.7.1. Képernyő tartalom tárolása és visszaolvasása

Az aktuális képernyőtartalmat háttértárolóra is kitehetjük a **GS**AV utasítással.

Alakja:

!GSAV "n",d

ahol *n* a kiírandó kép azonosítója (egy karaktersorozat, mint a programnév!),

d a háttértároló egység száma (1 esetben szalag, *d*=8-15 esetben lemez).

A képtartalom háttértárolóról történő visszahozására szolgál a RECALL utasítás.

Alakja:

IRECALL "n",d

ahol *n* és a *d* paraméterek megegyeznek a GSAV utasításnál elmondottakkal.

Példa: A két utasítás illusztrálására tekintsük az alábbi színek megadó programot.

```
10  !GCLEAR
20  !GRAPHICS 1
30  FOR T=0 TO 14
40  !LCOL T
50  !FILL T*20,0 TO T*20+20,199
60  NEXT
70  !GSAV "RAJZ-SZINEK", 8
80  GET A$:IF A$="" THEN 80
90  !GCLEAR:!GRAPHICS 1
100 IRECALL"RAJZ-SZINEK",8
```

A program a képernyőre kirajzolt képet elteszi mágneslemezre, majd egy tetszésszerű karakter leütése után a képernyőt kitisztázza és visszahozza lemezről és megjeleníti.

Amennyiben a képet szalagra akarjuk kitenni, úgy a következőket kell tennünk:

1. Az egységszámot 8-ról 1-re kell cserélni a 70-es és 100 -as utasításokban.
2. Be kell tenni egy kazettát a magnetofonba, és a rajzolás befejezése után be kell a PLAY és RECORD gombokat nyomni.
3. A felvétel befejezése után a képernyőtartalom újra megjelenik. Ekkor a szalagot vissza kell tekerni a felvétel elejére.
4. Meg kell nyomni egy tetszésszerű billentyűt a gépen /GET...!/.
5. Le kell nyomni a PLAY billentyűt a magnetofonon.
6. A beolvasás után a színeket tartalmazó képernyő újra megjelenik.

Megjegyzés: a 8K grafikus képernyőtartalom kiírásához összesen 37 blokk szükséges, a SUPERGRAFIK tehát összesen 5 további blokkot foglal a 8K képernyőtartalomhoz.

1.7.2. Képernyőtartalom kiírása nyomtatón

Grafikus képernyőtartalom kilistázását végzi a HARD utasítás.

Alakja:

!HARD n

ahol n a printer egységszám (4 vagy 5).

Eredménye: A grafikus képernyőtartalmat kiírja a nyomtatóra.

Megjegyzés: Az utasítás használata előtt az OPEN utasítást ki kell adni a printerre.

Példa: Az aktuális képernyőtartalom kiírásához bárhova a programba tegyük be a következő utasításokat:

```
OPEN 4,4
!HARD 4
```

Eredmény: az aktuális grafikus képernyőt kinyomtatja.

1.7.3. Egy képernyő pont ellenőrzése

Egy képernyő pont tesztelését végzi a TEST utasítás. A cél annak a kérdésnek az eldöntésére, hogy az aktuális képernyőn a vizsgált pontban történt-e rajzolás vagy sem.

Alakja:

```
!TEST x,y,v
```

ahol x,y a vizsgált pont koordinátái,

v a változónév, ahova a tesztelés eredményének értéke kerül:

$v=0$, ha nincs a pont kirajzolva,

$v=1$ egyébként.

1.7.4. Képernyőtartalmak cserélése

Lehetőség van grafikus képernyőtartalmak változtatására is úgy, hogy egy előbbit is ki tudunk rajzoltatni.

Alakja:

!TRANS

Mondanivalónk alaposabb megértéséhez futtassuk le és analizáljuk az alábbi programot.

```
10  !BCOL 2
20  !LCOL 0
30  POKE 53280,2
40  !GCLEAR
50  !GRAPHICS 5
60  !CIRCLE 100,100,20,50
70  !TRANS
80  !GCLEAR
90  !CIRCLE 100,100,50,20
100 !GRAPHICS 1
110 FOR T=1 TO 50
120 !TRANS
130 NEXT T
```

1.7.5. Alakzatok kirajzolása

Egyenes részekből összeállított tetszésszerű alaku geometriai ábrák kirajzolását végzi a DISP utasítás.

Alakja:

!DISP p,x,y,s

ahol p rajzolási paraméter:

p=0 esetén az s alakzat kitörölődik,

p=1 esetén az s alakzat kirajzolódik,

x,y a kirajzolandó alakzat első elemének koordinátái,

s a kirajzolandó alakzatot megadó karakterlánc.

Példa:

A következő program egy háromszöget rajzol ki a képernyőre, majd egy billentyű lenyomása után eltünteti

```

10  !BCOL 2
20  !LCOL 0
30  A$='X ]0]0]0]0]0]0]0]XQXQ ]XQ ]X
      Q ]XQ ]XQ ]X ||X ||X ||X ||X ||X ||X ||X ||X ||X ||X "
40  !GCLEAR
50  !GRAPHICS 1
60  !DISP 1,100,100,A$
70  GET B$:IF B$=""GO TO 70
80  !DISP 0,100,100,A$

```

1.7.6. Szöveg kiírása grafikus képernyőre

Szöveget a képernyőre kiírni a **TEXT** utasítással lehet.

Alakja:

```
!TEXT "s",x,y,p/,c/
```

ahol s a kiírandó szöveg,

x, y a kiíratandó szöveg kezdetének koordinátái,

p paraméter értékei:

$p=0$ esetben a szöveg kiíródik (nagybetű/grafika),

$p=1$ esetben a szöveg kiíródik (nagy/kisbetű),

c opcionális paraméter, a szöveg színe.

Példa:

Az 1.7.5. pontban közölt programot bővítjük a következő utasításokkal:

```
55 !TEXT "EZ EGY HAROMSZOG",20,20,0
75 !TEXT"          VOLT!!!".20.20.0
```

1.7.7. Supergrafika kikapcsolása

A SUPERGRAPHIK 64 szalagos változatának kikapcsolását végzi az OFF utasítás.

Alakja:

!OFF

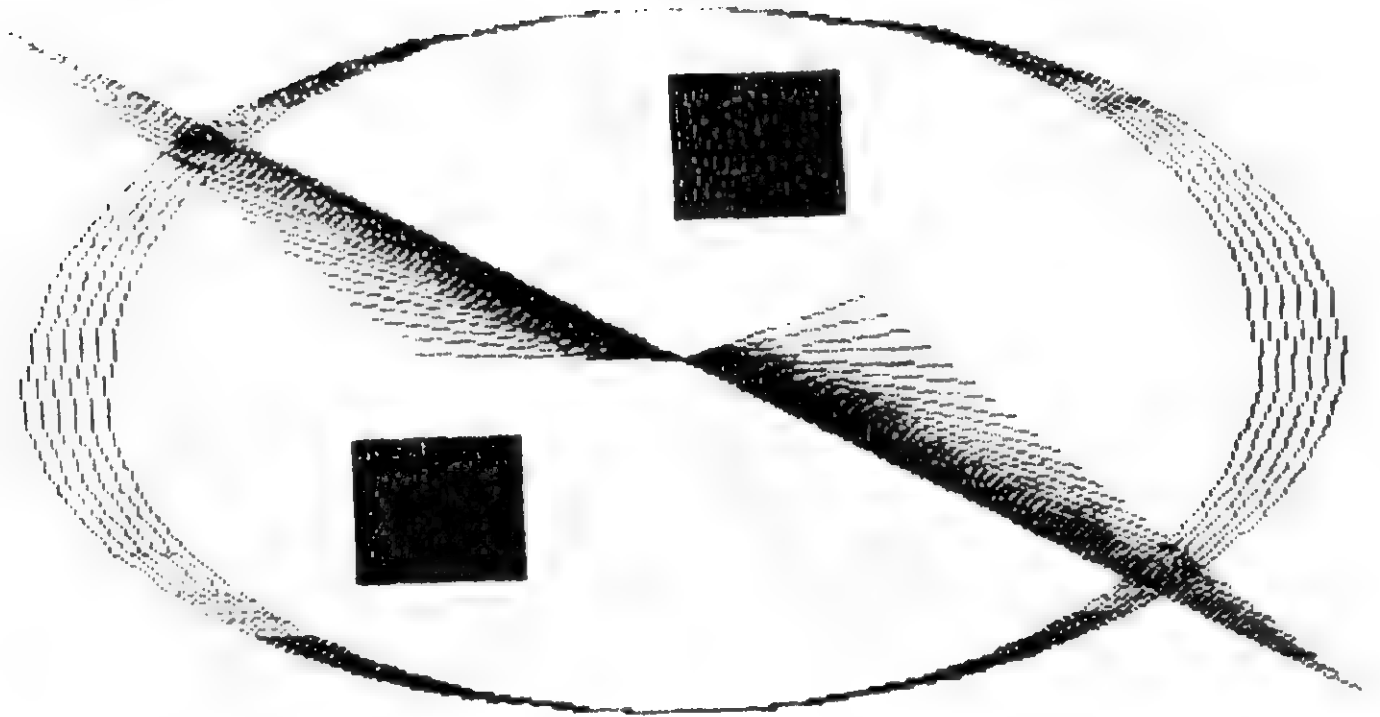
1.8. Példa program

Az alábbiakban bemutatunk egy programot, amely a rendszer utasításából többet használ, és érzékelteti a sokoldalu lehetőségeket .

```
10 POKE53280,1
20 !BCOL 1
30 !LCOL 6
40 !GCLEAR
50 !GRAPHICS 1
60 FORT=160 T0140 STEP-4
70 !CIRCLE160,100,T,100,90,270
80 !CIRCLE160,100,T-2,100,270,360
90 !CIRCLE160,100,T-2,100,0,90
100 NEXT T
110 !LCOL 2
120 FORT=1T0100 STEP5
130 !MOVE160,100
140 !DRAWT,T
150 NEXT
160 FORT=319T0200 STEP-5
170 !MOVE160,100
180 !DRAWT,T-120
190 NEXT T
200 FORT=0T015
```

```
210 !LCOL T
220 !FILL8Ø, 12ØT012Ø, 16Ø
230 !FILL16Ø, 2ØT02ØØ, 6Ø
240 NEXT
250 !GCLEAR
260 !GRAPHICS 1
270 !BCOLØ:POKE5328Ø, Ø
280 !LCOL1
290 FORT=13ØT018Ø
300 !LINET, 2ØØ-T/2T0 2ØØ-T/3, T
310 !LINET, 2ØØ-T/2T0 2ØØ-T/4, T
320 !LINET, 2ØØ-T/3T0 2ØØ-T/4, T
330 !CLINET, 2ØØ-T/2T0 2ØØ-T/3, T
340 !CLINET, 2ØØ-T/2T0 2ØØ-T/4, T
350 !CLINET, 2ØØ-T/3T0 2ØØ-T/4, T
360 NEXT
370 !GCLEAR
380 !GRAPHICS 1
390 POKE5328Ø, 2: !BCOL2: !LCOL1
400 FORT=8ØT02ØØ
410 !MOVE16Ø, 1ØØ
420 !DRAW T/2, T/2+INT(J)
430 !CLINE16Ø, 1ØØT0T/2, T/2+INT(J)
440 J=J+Ø.4
450 NEXT
460 !GRAPHICS Ø
470 END
```

A program egyik alakzatának rajzát alább közöljük:



2. Lemezes változat

A SUPERGRAPHIK 64 . legutóbb megjelent lemezes változata némileg eltér a korábbi, szalagos változattól.

Az utasítások között átfedés van. A lemezes változat elsősorban azal több a szalagos változatnál, hogy

- a zenei lehetőségekkel is kibővítették,
- létezik egy 80x50-es grafikus képernyő változata is.

2.1. Rendszer aktivizálása

A SUPERGRAPHIK lemezes változatát a következőképpen töltjük be a gépbe:

1. Bekapcsoljuk a lemezegységet (és természetesen a gépet is).
2. Begépeljük a következőket:

LOAD "*",8 /RETURN

A gép Uzenete néhány másodperces lemezforgás után / 3 mp/:

READY

3. Begépeljük a

RUN / RETURN

parancsot.

A floppy működésbe kezd, majd körülbelül 22-23 mp után a képernyő fekete lesz és rózsaszín betűkkel a következő Uzenetet írja ki:

SUPERGRAPHIK CBM 64

C 1983 BY DATA BECKER

Ezután használhatjuk a rendszer utasításait. A továbbiakban összefoglaljuk a különböző utasításokat. Részletes leírást a kézikönyvben találunk.

2.2. Zenei utasítások

A zenei utasítások köre nem túl széles és a következőkből áll:

Hangerő:

$$\text{VOLUME} = v$$

A SIMONS' BASIC VOL utasításához hasonlóan a hangerő beállítására alkalmas. A v értéke 0 és 15 közötti érték lehet. Nullánál "kikapcsol", 15 a legerősebb hang.

2.2.1. Hangjellemzők megadása

A hangjellemzők megadását a SOUND utasítással érhetjük el. Alakja:

$$\text{SOUND } s, w, a, d, p, r, (f(p))$$

- ahol
- s a hang megadása /1., 2 vagy 3. hangra vonatkozik az utasítás/,
 - w a hullámformát rögzíti /háromszög, négyszög, fűrészfog, zaj/,
 - a felfutás nulláról maximumra /attack/,
 - d lefutás átlagszintre /decay/,
 - p átlagszint ideje /sustain/,
 - r lefutás átlagszintről nullára (release),
 - $f(p)$ szűrő paraméter.

2.2.2. Szűrő beállítása

A szűrő paramétereinek beállítására szolgál a `FILTER` utasítás.

Alakja:

`FILTER a (,b,c)`

- ahol
- a a szűrő jellemző megadására szolgáló paraméter (0-7 közötti érték),
 - b szűrő határfrekvencia (0-2047 közötti érték),
 - c szűrő rezonancia frekvencia (0-15 közötti érték).

2.2.3. Hang megadása és lejátszása

A hang lejátszására szolgál a `TUNE` utasítás.

Alakja:

`TUNE s,l,n,o (,v)`

- ahol
- s a megszólaltatandó hang (1., 2. vagy harmadik hang),
 - l a hang hosszúsága 1/60-ad másodpercekben (0-65535 közötti érték),
 - n a megszólaltatandó hang magassága (értékei 1-12 között):
1:c, 2:cs, 3:d, 4:ds, 5:e
6:f, 7:fs, 8:g, 9:gs, 10:a
11:ais, 12:h
 - o az oktáv megadására szolgáló paraméter (0-7 közötti érték),
 - v a hangolásra szolgáló paraméter (értékei 0 és 255 közötti érték lehet).

Példaképpen tekintsük a következő programot:

```
10  VOLUME=15
20  SOUND 1,3,10,15,1,15,1
30  FILTER 1,1000,15
40  FOR J=0 TO 7
50  FOR I=1 TO 12
60  TUNE 1,50,1,J
70  NEXT I
80  NEXT J
```

A fenti program lejátsza a C64 gép minden hangját – összesen 96 hangot – a félhangokat is beleértve.

2.3. 80x50-es grafika

A 80x50-es grafika esetén az x és y értékei a következők lehetnek:

$$0 \leq x \leq 79$$

$$0 \leq y \leq 49$$

A képernyő 0,0 pontja a bal felső sarokban van. A fenti számokból látszik, hogy minden egyes betűhely négy részre van felbontva.

A grafika előnye, hogy a normál karaktereket minden további nélkül használhatjuk.

2.4. Rendszer utasításai

Az alábbiakban összefoglaljuk a rendszer utasításait. Részletes leírást a rendszerhez adott kézikönyvben találunk.

2.4.1. Rajzoló utasítások

PLOT x,y

x,y koordinátájú pontot megjelenít.

PLOT x_1,y_1 TO x_2,y_2

Szakasz rajzolása az x_1,y_1 pontból az x_2,y_2 pontba.

PLOT TO x_2,y_2

Az aktuális grafikus kurzor állásából az x_2,y_2 pontig egy egyenest húz.

CIRCLE x_1,y_1,x_2,y_2

Kör, ellipszis rajzolására szolgáló utasítás, paraméterei:

x_1,y_1 középpont,

x_2 vízszintes sugár,

y_2 függőleges sugár.

PAINT v FROM x,y

Alakzatok megszerkesztésére és rajzolására szolgáló utasítás, kezdete az x,y koordinátákkal megadott pont:

$v = 0$ felfelé egy pontnyit

$v = 1$ jobbra egy pontnyit

$v = 2$ lefelé egy pontnyit

$v = 3$ balra egy pontnyit.

PAINT v

A grafikus kurzor aktuális állásától kezdve rajzolja ki a kívánt alakzatot (v értékei mint a fenti utasításnál).

FRAME $d, x1, y1$ TO $x2, y2$

Egy d vastagságú rámát rajzol az $x1, y1$ bal felső és $x2, y2$ jobb alsó pontokkal megadott téglalap körületére.

FILL $x1, y1$ TO $x2, y2$

Adott színnel befestett téglalapot rajzol, amely bal felső csucának koordinátái $x1, y1$, jobb alsó csucának koordinátái $x2, y2$.

TEXT s, x, y, m

Az x, y koordinátákkal megadott ponttól kezdve kiírja az s szöveget az m móddal:

$m = 0$ nagybetűk/grafikus jelek

$m = 1$ nagybetűk/kisbetűk.

2.4.2. Színek megadása

FCOL f

Képernyő keretszínének megadása

$/f = 0 - 15/$.

BCOL f

Háttérszín megadása

$/f = 0 - 15/$.

SCOL

Rajzolósi szín megadása.

PCOL

Pont szín megadása.

2.4.3. Input-output utasítások

GSAVE $n, "f", g$

Az f nevű állományba helyezi

lemezre $/g=8/$ vagy szalagra

$/g=1/$ a képernyőn lévő rajzot.

GLOAD *n, "f", g*

HCOPY

Az *f* nevű grafikus képernyőt betölti és megjeleníti a *g* egységről. Különböző formáival a következő nyomtatókra rajzolhatjuk ki a képernyőtartalmat:

- SEICOSHA GP 100VC
- EPSON FX-80 /C 1526/
- SEICOSHA GP-700M /8 színű printer/
- fentiek valamelyikével kompatibilis printer.

2.4.4. Szellem grafika

SREAD *s*

Az *s* karaktertípusú változóba elhelyezi a szellem megrajzolásához szükséges adatokat /63 byte gyors olvasása/.

SDEFINE

Szellem megadásához szükséges paramétereknek értékadás.

SMODE *s, m, f*

Mód megadása, a paraméterek:

- s* a szellem sorszáma
- m* nagyítás és prioritás megadás
- f* szín /színek/ megadása.

SSET *s, x, y*

Hely megadására szolgáló utasítás:

- s* a szellem sorszáma
- x, y* a szellem bal felső sarkának koordinátái.

SSET $s, x_1, y_1, TO\ x_2, y_2, p$

Szellem mozgatására szolgáló utasítás:

s sorszámú szellemet az x_1, y_1 pontból az x_2, y_2 pontba p sebességgel ($1/60$ mp egy p értéke, max. érték: $p=256$).

SSET $s\ TO\ x, y, p$

(2)

Mint a fenti utasítás, de a mozgatás a szellem aktuális helyéről történik.

SWAIT s

A program futása az s sorszámú szellem mozgásának végéig áll.

IF par THEN...

Programelágaztató utasítás

RETURN

Megszakítás befejeződik.

2.4.5. Egyéb utasítások

GMODE s, m

Rajzolás módjának megadására szolgáló utasítás.

s értékei

értelmezése

0

1-es oldal látszik, utasítás erre vonatkozik

1

1-es oldal látszik, utasítás 2-re vonatkozik

2

2-es oldal látszik, utasítás 2-re vonatkozik

3

2-es oldal látszik, utasítás 1-re vonatkozik

m	érték	értelmezése
	0	LGR látszik, utasítás erre vonatkozik
	1	HRG látszik, utasítás erre vonatkozik
	2	MC látszik, utasítás erre vonatkozik
	3	HRG látszik, utasítás LGR-re vonatkozik
	4	MC látszik, utasítás LGR-re vonatkozik
	5	LGR látszik, utasítás HRG-re vonatkozik
	6	LGR látszik, utasítás MC-re vonatkozik
		ahol LGR 80x50-es grafika HRG 320x200-as grafika MC 160x200-as grafika .
GCLEAR		Grafikus képernyő kitörlése.
GMOVE		Grafikus ábrák jobbra /balra/ mozgására szolgáló utasítás.
ROT⇒		A PAINT utasítással kirajzolandó alakzatok forgatására szolgáló utasítás /r 0 és 63 közötti érték, azaz 1 r értéke 5.625 fok/
SIZE = s		A PAINT utasítással megadott alakzatok nagyítását végzi az utasítás, /s értéke 0-255 között változhat, s=0 esetén nincs nagyítás/.
INVERS		A rajz inverzét adja, azaz azon pontok, amelyek ki voltak "rajzolva", nem lesznek a továbbiakban és viszont!

C64 AIDS

1. Bevezetés

A C64 AIDS a Print-Technik /WIEN/ terméke. Az egyéb segítőkártyához hasonlóan a programozást, programbelövést segíti. Összesen 16 utasításból áll, ezek közül az egyik a HELP, amely a rendelkezésre álló parancsokat listázza ki. A kártya hasonló segítséget ad, mint a HELP PLUS, a VIC 1212 és a SIMON'S BASIC parancsai. Összehasonlítva a fentiekkel a következőket állapíthatjuk meg:

- a HELP PLUS és a SIMON'S BASIC megfelelő utasításai kényelmesebbek, többet nyújtanak,
- a VIC 1212 egyrészt több, másrészt kevesebb, mint a C64 AIDS, de nem sokkal tér el tőle.

A továbbiakban összefoglaljuk a programrendszer utasításait.

2. Programozást segítő parancsok

2.1. Programok összemásolása

Programok összemásolását végzi az **APPEND** parancs.

Alakja:

APPEND "f"

ahol **f** a másolandó programnév.

Hatása: a gépben levő program végétől kezdve hozzámásolja a lemezen levő **f** nevű programot.

Igy a másolásnál biztosítani kell, hogy a másolandó program kezdő sorszáma nagyobb legyen a gépben levő program legnagyobb sorszámanál.

2.2. Programsorszámok automatikus kiírása

A programozást segíti az **AUTO** automatikus sorszámozást végző utasítás.

Alakja:

AUTO s,i

ahol **s** a sorszámozás kezdete,

i az egyes programsorok közötti növekmény.

2.3. Program egyes részeinek eltörlése

A program egyes részeit törli ki a **DELETE** parancs.

Alakja:

DELETE n,m

ahol n az eltörlendő első programsor sorszáma,
m az eltörlendő utolsó sor sorszáma.

Hatása: Az n. utasítás sorszámtól az m.-ig eltörli az utasításokat -
n-t és m-et is beleértve.

2.4. Program újracsorjámozása

Egy program újracsorjámozását végzi a RENUMBER parancs.

Alakja:

RENUMBER s,i,n,m

ahol s az átsorjámozandó programrész első utasításának új sor-
száma,
i az átsorjámozandó programrész utasítás sorszámai közötti
különbség,
n az átsorjámozandó programrész első utasításának sorszáma,
m az átsorjámozandó programrész utolsó utasításának sorszáma.

2.5. Karakterek sokszorozása

Karakterek sokszorozását végzi el a REPEAT parancs.

Alakja:

REPEAT

Hatása: A parancs kiadása után egy tetszőszerinti billentyűt lenyomva
tartva addig írja ki a gép, ameddig fel nem engedjük.

3. Programbelövés segítése

3.1. Változók értékének kiírása

Egy program változói értékeinek kiírását végzi el a **DUMP** parancs.
Alakja:

DUMP i

ahol i utasítás paraméter a következő értékeket veheti fel:

%, \$, (, Ures

Hatása:

i = %	esetén kilistázza az egész típusú változók nevét és aktuális értékét,
i = \$	esetén kilistázza a karakter típusú változók nevét és aktuális értékét,
i = (esetén kilistázza a tömbök neveit és megadja a dimenziójukat,
i = Ures	esetén a fentiek mindegyikét megjeleníti a képernyőn.

3.2. Program stringjeinek kicserélése

Egy program stringjeinek kicserélését végzi az **EDIT** parancs.
Alakja:

EDIT /s1,s2/ (n-m, Q)

ahol	s1,s2	karakterláncok,
	s1	amit ki akarunk cserélni,
	s2	amire cseréljük s1-et,
	n	opcionális paraméter, a figyelembe veendő programrész első utasításának sorszáma,
	m	opcionális paraméter, a figyelembe veendő programrész utolsó utasításának sorszáma,
	Q	ha szerepel, akkor minden s1 megtalálása esetén megkérdezi, hogy cseréljen-e, vagy sem.

Hatása: Az n-edik sorszámu sortól az m-edik sorszámu sorig s1-et s2-re cseréli. Ha Q szerepel, akkor csak "Y" válasz esetén cserél "N" válasz esetén továbbmegy.

3.3. String megkeresése

Egy programban előforduló stringek megkeresését végzi a `FIND` parancs.

Alakja:

`FIND / s / n,m`

ahol	s	a megkeresendő string,
	n	az első programsor sorszáma, ahonnan a keresés kezdődik,
	m	az utolsó programsor sorszáma, ameddig a keresés még tart.

Hatása: n-m közötti programsorokban megkeresi az s stringet, és ki-listázza azon sorokat, melyekben ez a string előfordul.

3.4. Programfutás nyomonkövetése

A program futásának nyomonkövetését végzi a `TRACE` utasítás.

Alakja:

TRACE W

ahol W opcionális paraméter.

Hatása:

- Ha a W paraméter nincs kitéve, akkor a program futása lelassul,
- Ha W-t is begépelünk, akkor az utolsó négy végrehajtott utasítás sorszámát írja ki a gép a képernyő jobb felső sarkába.

A TRACE utasítást az OFF utasítással lehet hatástalanítani.

Alakja:

OFF

A program lépésenkénti végrehajtását végezhetjük a STEP utasítással.

Alakja:

STEP W

ahol W opcionális paraméter.

Hatása:

- Ha W paramétert nem használjuk, a programot lépésenként hajtja végre és minden utasítás után megáll. Tovább indítani a SHIFT gombbal lehet.
- Ha a W paraméter is ki van téve, úgy a fentiek mellett az utolsó 4 végrehajtott utasítás sorszámát is kiírja a képernyő jobb felső sarkába.

4. Konstansok konvertálása

Decimális szám konvertálását kettes és tizenhatos számrendszerbeli szám-
ba a DEC függvény végzi.

Alakja:

DEC n1 n2 n3 n4 n5

ahol ni decimális számjegyek.

Példa

DEC 64444
=HFBBC
= B 11111011 10111100

Tizenhatos számrendszerbeli számot tizesbe és kettesbe a HEX függvény
konvertál át.

Alakja:

HEX h1 h2 h3 h4

ahol hi tizenhatos számrendszerbeli számjegy.

Példa

HEX AAAD
=43693
=B10101010 10101110

5. Egyéb parancsok

5.1. Rendszer kikapcsolása

A rendszer hatását a **KILL** paranccsal szüntethetjük meg.

Alakja:

KILL

5.2. Eltörölt program visszaállítása

A **NEW** paranccsal eltörölt programot visszaállítani az **OLD** paranccsal lehet.

Alakja:

OLD

I. F Ü G G E L É K

SIMONS' BASIC

hibaÜzenetek

Üzenet	jelentése
?BAD MODE	Egy parancs paramétere nem megegyezett típusu.
?NOT HEX CHARACTER	Nem hexadecimális számot akartunk decimálissá alakítani.
?NOT BINARY CHARACTER	Nem kettes számrendszerbeli számot akartunk decimálissá alakítani.
?UNTIL WITHOUT REPEAT	Egy UNTIL utasítást használtunk olyan helyen, ahol előzőleg nem volt REPEAT utasítás.
?END LOOP WITHOUT LOOP	Egy END LOOP utasítást használtunk LOOP utasítás nélkül.
?END PROC WITHOUT EXEC	Egy END PROC utasítást használtunk PROC utasítás nélkül.
?PROC NOT EXIST	Egy olyan eljárást kíséreltünk meg behívni, amelyik nem létezik.
?NOT ENOUGH LINES	Nincs elég sor megadva egy szellem megszerkesztéséhez.
?BAD CHAR FOR A MOB	A szellem megadásánál a DESIGN utasításban a paraméterek nem esnek a megadott intervallumba.
?INSERT PARAMETER TOO LARGE	Az INSERT parancs esetén fordul elő akkor, ha a hossz megadása nem megfelelő nagyobb, mint a karakterlánc hossza.

?CREATED STRING TOO LONG

Az **INSERT** utasítással létrejött karakterlánc hossza meghaladja a megengedett 255 karaktert.

BAD MODE

A **MOVE** utasítással nagyobb területet jelöltünk ki, mint amennyi a képernyőn elfér.

II. F Ü G G E L É K

HELP PLUS parancsai

Programozás, listázás segítése

#G a,b

Automatikus sorszámkiírás

a kezdő sorszám

b növekmény.

*

Szabad memóriahely kiírása.

#L n

Program listázása képernyőnként

n első listázandó sor
száma.

#R a,b

Program átsorszámozása

a kezdő sorszám

b növekmény.

#C a

Program sűritése

a az egy sorban elhelyezhető karakterek
száma.

#A "n", d

Gépben lévő programhoz másik programrész másolása külső tárolóról

n másolandó programnév

d egységszám, ahol a másolandó program van.

F s

Egy programrészlet megkeresése

s a programban előforduló tetszőszerinti karakter sorozat.

#D n1-n2

Programsorok törlése

n1-től n2-ig eltörli a sorokat.

Programbelövés segítése

#H

Utolsónak végrehajtott program sor kiírása megszakításkor /befejezéskor/.

#T

Program végrehajtásának folyamatos soronkövetése.

#S

Program végrehajtása utasításonként.

#E

#T vagy #S parancs befejezése.

#U

Nem megengedett programhivatkozások /GO TO, GOSUB stb./ ellenőrzése.

#V

Egyszerű programváltozók tartalmának kiírása képernyőre.

#M

Egy program indexes változóinak értékét kiíró parancs.

!# k1k2k3k4

Tizenhatos számrendszerbeli szám konvertálása tizesbe

ki tizenhatos számrendszerbeli számjegy.

!\$ n1n2n3n4n5

Tizes számrendszerbeli szám konvertálása tizenhatosba

ni tizes számrendszerbeli számjegy.

Egyéb parancsok

#B

NEW paranccsal eltörölt program visszaállítása.

(d

OPEN és CMD parancs együttes kiadására szolgáló parancs

d egységszám.

)

Fenti paranccsal megnyitott file lezárására szolgáló parancs.

f d

Lemez egységszám megváltoztatására szolgáló parancs

d új egységszám.

#K

HELP PLUS befejezésére szolgáló parancs.

Disassembler parancsok

]	Disassembler aktivizálására szolgáló parancs.
+	Cimváltoztatásra szolgáló parancs.
/	Egy byte tartalmának megváltoztatására szolgáló parancs.
[Nyomtatón való listázás /disassemblálás/ parancsa.
>	Gép memóriájának disassemblálását kijelölő parancs.
<	Floppy egység memóriájának disassemblálását kijelölő parancs.
RETURN	Cimet a listázásnál eggyel növeli .
↑	Cimet a listázásnál eggyel csökkenti.
-	A beállított cimtől folytatódik a disassemblálás.
space	Folyamatos disassemblálás.
CTRL	Kiírás lassítása.
STOP	Kiírás megállítása.

SHIFT

Kiírás átmeneti megállítása.

SHIFT/LOCK

Benyomva a listázás átmenetileg áll, elengedve továbbindul.

*

Gépi kódu program elindítása a disassembláció aktiv állapotában.

@

Memóriaterület átmásolása.

=

Disassemblálás kikapcsolása.

Disk parancsok

/ n

Programbetöltés lemezről
n betöltendő program neve.

% n

Programbetöltés LOAD "n", 8, 1
alakban
n betöltendő programnév.

↑ n

Programbetöltés és futási parancs
n a betöltendő és futtatandó program neve.

← n

Programkiírás lemezre
n a kiírandó program neve.

< n

Programkiírás ellenőrzése
(VERIFY)
n az ellenőrzendő program
neve.

§

Lemezdirektori kiírása képernyőre.

@

Disk státusz kiírása.

>

Parancscsatornáról olvasás.

Assembler fordító parancsai

[

Fordító aktivizálása.

III. F Ü G G E L É K

SIMON'S BASIC utasításai és parancsai

Rajzoló utasítások

HIRES x, y

Átkapcsolás grafikus módra

x rajzolás színe

y háttérszín.

MULTI x, y, z

Szines grafikus módra való
átkapcsolás

x 1. rajzolási szín

y 2. rajzolási szín

z 3. rajzolási szín .

LOW COL p, q, r

További színek megadása

p 1. rajzolási szín

q 2. rajzolási szín

r 3. rajzolási szín.

HI COL

Visszatérés az eredeti rajzo-
lási színre.

Geometriai alakzatok megjelenítése

PLOT x, y, p

Egy pont megjelenítése a gra-
fikus képernyőn

x, y pont koordinátái

p rajzolási típus.

LINE x_1, y_1, x_2, y_2, p

Egy egyenes szakasz kirajzolása

x_1, y_1 szakasz kezdőpontjának koordinátái

x_2, y_2 szakasz végpontjának koordinátái

p rajzolási típus.

REC x_1, y_1, x_2, y_2, p

Téglalap kirajzolása

x_1, y_1 téglalap bal felső sarkának koordinátái

x_2, y_2 téglalap jobb alsó sarkának koordinátái

p rajzolási típus.

CIRCLE x_1, y_1, x_2, y_2, p

Kör, ellipszis kirajzolása

x_1, y_1 alakzat középpontjának koordinátái

x_2 alakzat félszélessége

y_2 alakzat félmagassága

p rajzolási típus.

ARC $x_1, y_1, a, b, c, x_2, y_2, p$

CIRCLE utasítással kirajzolható alakzat egy részének kirajzolására szolgáló utasítás

x_1, y_1 u.a., mint a CIRCLE utasításnál

a rajzolás "kezdetének" szöge

b rajzolás végének szöge

c lépésköz

x_2, y_2 u.a. mint a CIRCLE utasításnál

p rajzolási típus.

ANGL x_1, y_1, a, x_2, y_2, p

CIRCLE utasítással kirajzolható alakzat hurjainak kirajzolása

x_1, y_1 lásd CIRCLE utasítás

a függőleges "sugárral" bezárt szög

x_2, y_2 u.a. mint a CIRCLE utasításnál.

DRAW "nnn...9", x, y, p

Grafikus kurzor mozgatása rajzolással vagy anélkül

n mozgatás iránya és rajzolás/nem rajzolás megadása

x, y rajzolás kezdetének koordinátái

p rajzolási típus.

RCT s, m

DRAW utasítással megadott alakzat kirajzolása

s forgatás szöge

m méret növelési faktor.

Alakzatok színezése

PAINT x, y, p

Egy zárt alakzat kifestése

x, y az alakzat egy belső pontjának koordinátái

p rajzolási típus.

BLOCK x_1, y_1, x_2, y_2, p

Szines téglalap kirajzolása

x_1, y_1	téglalap bal felső sarkának koordinátái
x_2, y_2	téglalap jobb alsó sarkának koordinátái
p	rajzolási típus.

Rajzok feliratozása

CHAR x, y, c, p, s

Egy karakter kiírása a grafikus képernyőre

x, y	karakter kezdetének koordinátái
c	kirajzolandó karakter POKE kódja
p	rajzolási típus
s	kirajzolandó karakter mérete.

TEXT $x, y, "/t/b", p, s, i$

Egy szöveg kiírása a grafikus képernyőre

x, y	kiírandó szöveg kezdetének koordinátái
t	alsó/felsőállás vezérlése
b	kiírandó karaktersorozat
p	rajzolási típus
s	karakter mérete
i	üres cellák száma a karakterek kezdete között.

További grafikus utasítások

CSET 1

Karakterkészletek váltása, utolsó grafikus képernyő megjelenítése

i=0 felső állás/grafikus karakterkészlet

i=1 alsó/felső állás

i=2 legutolsó grafikus képernyő kirajzolása.

TEST (x,y)

Megadott pont állapotának ellenőrzése

x,y megadott pont koordinátái.

BASIC utasítások kiterjesztése

IF...THEN...ELSE...

Feltételes utasítás kiterjesztése.

RCOMP...ELSE...

Az utasítást megelőző IF...THEN ...ELSE utasításra való hivatkozás.

REPEAT...UNTIL L

Ciklusutasítás, a REPEAT UNTIL közötti utasításokat addig hajtja végre, ameddig L igaz nem lesz,
L logikai kifejezés.

LOOP...EXIT IF L END LOOP

A LOOP és EXIT közötti utasításokat addig hajtja végre, ameddig L értéke igaz nem lesz,
L logikai kifejezés.

c1 PROC n... END PROC

Zárt szubrutin,

c1 kezdőcímké

n szubrutin név

END PROC szubrutin végét jelzi.

c PROC n

Címke megadása

c utasításcímke

n szimbolikus név.

CALL n

Előző utasítással megadott helyre való ugrás

n szimbolikus név.

c LOCAL l

Helyi változók megadása

c utasításcímke

l helyi változók listája.

c GLOBAL l

Előző utasítással helyinek deklarált változókat visszaállítja eredeti értékekre

c utasításcímke

l globális változók listája.

CGOTO e

e aritmetikai kifejezés egész értékére történő ugrás.

RESET c

DATA utasítás aktuális pointerének megváltoztatása

c egy DATA utasítás címkéje.

Írás-olvasás utasításai

CENTRE "s"

Képernyőre kiírandó szöveg középre
helyezése

s kiírandó karakterlánc.

USE "p.q",v:PRINT

v változó értékének kiírása p egész
és q tizedes jegyre

p és q jelek sorozata

USE "s1p.qs2",v:PRINT

v változó értékének kiírása s1 "kez-
dőszöveggel", p egész q tizedes
jegyre s2 befejező szöveggel

p és q jelek sorozata

s1,s2 tetszőszerinti karakter -
lánc.

PRINT AT (p,q)"s"

p. oszlop q. elemétől kiírja az s
karakterláncot.

PRINT AT (p,q)v

p. oszlop q. elemétől kiírja a v
változó értékét.

FETCH "v",n,vn

vn változóba n karakter beolvasá-
sa a v vezérlő-karakterek megfe-
lelően.

INKEY

Numerikus értékű függvény, értéke a
lenyomott funkció billentyű száma
/1-16/vagy 0.

Egyéb utasítások

PAUSE "u",i

Programfutas felfuggesztése az u
karaktersorozat kiirásával i má-
sodpercre.

PAUSE i

Programfutas felfuggesztése i
másodpercre.

ON ERROR: GO TO c

1-5. és 10-23 sorszamu hibák
programból lekezelése

c	utasitascimke
ERRLN	hibas utasitas sor- szama
ERRN	hiba kodja.

NO ERROR

Fenti hibák lekezelése után a fu-
tas továbbinditása.

DISAPA:u

Program utasitassoranak elokészí-
tése titkositásra

u	titkositando uta- sitas.
---	-----------------------------

SECURE 0

Titkositas elvezesere szolgalo
parancs.

BASIC függvények kiterjesztése

MOD (a,b)

Modulus függvény: a moduló b.

DIV (a,b)

Egészrész osztás

a,b egész vagy valós
számok.

FRAC (k)

k aritmetikai kifejezés tizedes
része.

EXOR (a,b)

Kizáró vagy a és b számok
/bináris alakja/ esetén.

INSERT ("s1","s2",n)

Karakter sorozat beszúrása

s1 beszúrandó karak -
tersorozat
s2 karakterlánc, ami-
be a beszúrás tör -
ténik
n azon pozíció helye,
ahonnan a beszúrás
történik s2-ben.

INST ("s1","s2",n)

Karakterlánc végének megváltoz-
tatása

s2 karakterlánc, a végét
változtatjuk meg
s1 az s2 végére irandó
karakterlánc
n azon karakter pozíci-
ójának helye, ahon-
nan a változtatás tör-
ténik s2-ben.

PLACE ("s1","s2")

Karakterlánc helyének megkeresése

s1 keresendő karakterlánc

s2 karakterlánc, amiben keresünk.

DUP ("s",n)

Karakterlánc többszörözése

s többszörözendő karakterlánc

n többszörözés paramétere.

%b

Bináris szám konvertálása tizedesbe

b bináris szám.

%h

Tizenhatos számrendszerbeli szám konvertálása tizesbe

h hexadecimális szám.

PENX

Fényceruza aktuális helyének x koordinátája.

PENY

Fényceruza aktuális helyének y koordinátája.

POT (i)

Paddle potenciométerének aktuális állása

i berendezés száma /0 v. 1/.

JOY

Botkormány és a tüzelőgomb állásának meghatározása.

Képernyő műveletek

FLASH sz,s

Képernyő színeinek villogtatása

sz	szinkód
s	sebesség.

OFF

Képernyő villogtatásának befejezése.

BFLASH s,sz1,sz2

Képernyő keretének megadott színekkel történő villogtatása

s	sebesség
sz1,sz2	villogtatandó színek kódjai.

BFLASH 0

Keret színvillogtatásának befejezése.

FCHR a,b,c,d,e

Képernyő adott területének adott karakterekkel való feltöltése

a	kezdő sor sorszáma
b	kezdő oszlop oszlop - száma
c	terület szélessége karakterekben
d	terület "mélysége" karakterekben
e	megjelenítendő karakter POKE kódja.

FCOL a,b,c,d,e

Képernyő adott területén levő jelek színének megváltoztatása

- a-d megegyezik az FCHR utasításnál elmondottakkal
- e terület karaktereinek színe.

FILL a,b,c,d,e,f

Adott képernyő-terület adott színű karakterekkel való feltöltése

- a-e megegyezik az FCHR utasításnál elmondottal
- f területen megjelenített karakter szín kódja.

MOVE a,b,c,d,e,f

Képernyő aktuális tartalmának átmásolása a képernyő más területére

- a-d megegyezik az FCHR utasításnál elmondott paraméterekkel
- e az új terület bal felső sarkának sorindexe
- f az új terület bal felső sarkának oszlopindexe.

INV a,b,c,d

Karakterek inverz képének megjelenítési helyét jelöli ki

- a-d területet megadó paraméterek /lásd FCHR utasítás/.

LEFT i,a,b,c,d

Képernyő egy területének rollozása
balról jobbra

- i tekerés típusa
- a,b mozgatási terület bal felső sarkának oszlop ill. sor indexe
- c,d mozgatási terület jobb alsó sarkának oszlop ill. sor indexe

RIGHT i,a,b,c,d

Képernyő egy területének rollozása
jobbról balra

- i-d paramétereket a LEFT utasításnál ismertettük.

UP i,a,b,c,d

Képernyő egy területének rollozása
alulról felfelé

- i-d paramétereket a LEFT utasításnál ismertettük.

DOWN i,a,b,c,d

Képernyő egy területének rollozása
felülről lefelé

- i-d paramétereket a LEFT utasításnál ismertettük.

SCRSV 2,8,2,"n,S,W"

Képernyő aktuális tartalmának elhelyezése mágneslemezen

- n adatállomány neve.

SCRSV 1,1,1,"n,S,W"

Képernyő aktuális tartalmának elhelyezése mágnesszalagon

n adatállomány neve.

SCRLD 2,8,2,"n"

Képernyő adatainak visszaolvasása lemezről, majd megjelenítése

n adatállomány neve.

COPY

Karakteres vagy grafikus képernyő tartalom kiírása nyomtatón.

HRDCOPY

Karakteres képernyő tartalom kiírása nyomtatón.

LIN

Kurzor aktuális sorszámanak meghatározása.

Programírást segítő parancsok

AUTO s,n

Automatikus program-sorszámok generálása

s kezdő sorszám

n növekmény.

RENUMBER s,n

Gépben lévő program átsorszámozása

s új kezdősorszám

n növekmény.

PAGE n	Program listázása képernyőre részenként n egyszerre a képernyőre kiírandó sorok száma.
DELAY n	Program listázásának lassítása n listázás sebessége.
OPTION n	SIMON'S BASIC utasítások kiírása fordított /REVERSE/ módban /n=10 esetén/ n ≠ 10 és 0 és 255 közötti érték: normál listázás.
FIND c	c karakterlánc előfordulási helyének /helyeinek/ megkeresése.
<u>Szellem és karakter készítése</u>	
DESIGN a,b	Helyfoglalás egy szellem /sprite/ részére a típus b tároló első byte-jának címe.
@	Egyszínű, vagy színes szellem illetve saját karakter megszerkesztéséhez használatos séma.
CMOB s1,s2	Színek megadása multikolor módban s1,s2 szinkódok.

MOB SET a,b,c,d,e

Szellem paramétereinek megadása

- a szellem száma
- b blokkszám
- c szinkód
- d prioritás
- e típus kijelölése.

MMOB a,x1,y1,x2,y2,b,c

Szellem megjelenítése és mozgatása

- a szellem száma
- x1,x2 megjelenítés kezdetének koordinátái
- x2,y2 mozgítás végpontjának koordinátái
- b szellem mérete
- c mozgási sebesség.

RLOCMOB a,x,y,b,c

Képernyőn levő szellem mozgítása

- a szellem száma
- x,y mozgítás végpontjának koordinátái
- b,c megegyeznek a MMOB utasításnál elmondottakkal.

DETECT n

Alakzatok összeütközésének lekérdezését készíti elő

- n alakzat meghatározása.

IF CHECK (m1,m2)=0 THEN u

Összeűtközések lekérdezése

m1,m2 szellemek száma

u utasítás.

IF CHECK(0)=0 THEN u

Szellem és karakterek összeűtközésének lekérdezése.

MOB OFF

Megjelenített szellem kitörlése a képernyőről.

MEM

Karakterkészlet átmásolása ROM-ból RAM-ba.

DESIGN 2, %E000+ct8

Kicserélendő karakter megadása

c karakter POKE kódja.

Zenei utasítások

VOL n

Hangerő megadása.

WAVE h,b

Hang megadása

h hang száma

b hangjellemzők paraméterei.

ENVELOPE h,A,D,S,T

Burkoló görbe megadása

h hang száma

- A felfutás maximuma
- D lefutás egyenletes
erősségre
- S egyenletes erősség
ideje
- T lefutás nullára.

MUSIC n,"c"

A zene tempójának és hang -
jainak megadása

- n tempó paraméter
- c zenét megadó karak-
terlánc.

PLAY n

Zene lejátszása

- n lejátszás módja.

Disk parancsok

DISK,"művelet"

Disk műveletek megadása.

DIR "%....

Disk állományok neveinek kli-
rása.

További parancsok

OLD

NEW parancs hatástalanítása.

KEY s,"c"

Funkció billentyűk aktivizálása

s billentyű száma

c utasításnév.

DISPLAY

Funkció billentyű értékének ki -
írása.

IV. F Ü G G E L É K

SUPERGRAPHIK 64 szalagos változat utasításai

!GCLEAR

Grafikus képernyő kitörlése.

!GRAPHICS p

Grafikára való átváltás és típus-
kijelölés

p rajzolási paraméter.

!DISP p,x,y,s

Egyenesekből összeállított alak -
zat kirajzolása

p rajzolási paraméter

x,y rajzolás kezdetének
koordinátái

s kirajzolandó alakzatot
megadó karakterlánc.

!DOT x,y

Egy megadott koordinátájú pont
kirajzolása

x,y megjelenítendő pont ko-
ordinátái.

!CDOT x,y

Egy megadott koordinátájú pont tör-
lése

x,y kitörlendő pont koordiná-
tái

!TEST x,y

Egy megadott koordinátájú pont ál-
lapotának tesztelése

ILINE x_1, y_1 TO x_2, y_2

Egyenes szakasz rajzolása két koordináta pontokkal megadott pont között

x_1, y_1 kezdőpont koordinátái

x_2, y_2 végpont koordinátái.

ICLINE x_1, y_1 TO x_2, y_2

Egyenes szakasz törlése

x_1, y_1 kezdőpont koordinátái

x_2, y_2 végpont koordinátái.

IDLINE x_1, y_1 TO x_2, y_2

Egyenes szakasz kirajzolása pontozva a megadott pontok között

x_1, y_1 kezdőpont koordinátái

x_2, y_2 végpont koordinátái.

IFILL x_1, y_1 TO x_2, y_2

Téglalap kirajzolása és befestése

x_1, y_1 bal felső sarok koordinátái

x_2, y_2 jobb alsó sarok koordinátái.

ICLEAR x_1, y_1 TO x_2, y_2

FILL utasítással rajzolt alakzat törlése

x_1, y_1 bal felső sarok koordinátái

x_2, y_2 jobb alsó sarok koordinátái

IFRAME x_1, y_1 TO x_2, y_2

Két koordináta ponttal megadott téglalap oldalainak kirajzolása

x_1, y_1 bal felső sarok koordinátái

x_2, y_2 jobb alsó sarok koordinátái.

!CFRAME x_1, y_1 TO x_2, y_2

FRAME utasítással kirajzolt téglalap törlése

x_1, y_1 bal felső sarok koordinátái

x_2, y_2 jobb alsó sarok koordinátái.

!MOVE x, y

Grafikus kurzor mozgatása rajzolás nélkül a megadott pozícióba

x, y koordinátákkal megadott pontba.

!DRAW x, y

Egyenes szakasz rajzolása a kurzor aktuális pozíciójától kezdve a megadott pontig

x, y végpont koordinátái.

!GSAV "n", d

Aktuális grafikus képernyőtartalom elhelyezése mágneslemezre vagy mágnesszalagra

n állomány neve

d egységszám,

!RECALL "n", d

GSAV utasítással háttértárolóra klírt grafikus képernyőtartalom beolvasása és megjelenítése

n állomány neve

d egységszám.

!HARD n	Aktuális grafikus képernyőtartalom kiírása sornyomtatón n nyomtató száma.
!TRANS	A két grafikus képernyőtartalom közötti felcserélő utasítás.
!SPRITE n,v	Szellem adatainak megadása n szellem száma v szellem adatait tartalmazó változó.
!SMODE n,s,p	Szellem méretének és színének megadása n szellem száma s nagyság p szellem színe.
!SMOVE n,x,y	Szellem mozgatása n szellem száma x,y végpont koordinátái.
!BCOL n	Képernyő háttérszínét megadó utasítás n háttérszín kódja.
!LCOL n	Grafikus rajzolási szín megadása n rajzolási szín kódja.

!TEXT "s",x,y,p,/,c/

Szöveg kiírása /és eltörlése/ grafikus képernyőre /képernyőről/

s kiírandó szöveg
x,y kiírás kezdetének koordinátái
p karakterkészlet
c szöveg színe.

!CIRCLE x1,y1,x2,y2,/s1,s2/

Megadott középpontu, szélességű és magasságú kör /ellipszis/ kirajzolása

x1,y1 alakzat középpontjának koordinátái
x2 x irányu sugár hossza
y2 y irányu sugár hossza
s1 a rajzolás kezdőszöge fokokban
s2 a rajzolás "vég"-szöge fokokban.

!DCIRCLE x1,y1,x2,y2,/s1,s2/

Megadott középpontu, szélességű és magasságú kör /ellipszis/ kirajzolása pontozással

a paraméterek megegyeznek a CIRCLE utasításnál elmondottakkal.

!CCIRCLE x1,y1,x2,y2,/s1,s2/

Grafikus képernyőre kirajzolt kör /ellipszis/ eltörlése

a paraméterek megegyeznek a CIRCLE utasításnál elmondottakkal.

!OFF

Rendszer lekapcsolása.

V. F Ü G G E L É K

SUPERGRAPHIK 64 lemezes változat utasításai

Zenei utasítások

VOLUME = v

Hangerő beállítása.

SOUND s,w,a,d,p,r,(f(p))

Hangjellemzők megadása

- s hang sorszáma
- w hullámforma
- a felfutás nulláról maximumra
- d lefutás átlagszintre
- p átlagszint ideje
- r lefutás nullára
- f(p) szűrő.

FILTER a(b,c)

Szűrő paramétereinek beállítása

- a szűrő jellemző megadása
- b szűrő határfrekvencia
- c szűrő rezonancia frekvencia.

TUNE s,l,n,o,(v)

Hang lejátszása

- s hang sorszáma

l	hang hossza
n	magasság
o	oktáv szám
v	hangolási paraméter.

Rajzoló utasítások

PLOT x,y

Az x,y pont megjelenítése.

PLOT x1,y1 TO x2,y2

Szakasz kirajzolása

x1,y1 kezdőpont koordinátái

x2,y2 végpont koordinátái.

PLOT TO x,y

Egyenes rajzolása az x,y pontig.

CIRCLE x1,y1,x2,y2

Kör /ellipszis/ rajzolása

x1,y1 középpont koordinátái

x2 x irányu sugár hossza

y2 y irányu sugár hossza.

PAINT v FROM x,y

Egyenes szakaszokból összeállított alakzatok kirajzolása

v irány

x,y rajzolás kezdetének koordinátái.

PAINT v

Fenti alakzat kirajzolása a kurzor aktuális helyétől kezdve.

FRAME d,x1,y1 TO x2,y2

Egy téglalap alakú ráma rajzolása

d ráma vastagsága

x1,y1 bal felső sarokpont ko-
ordinátái

x2,y2 jobb alsó sarokpont ko-
ordinátái.

FILL x1,y1 TO x2,y2

Szines téglalap kirajzolása

x1,y1,x2,y2 megegyezik a
FRAME utasításnál elmondottak-
kal.

TEXT s,x,y,m

Szöveg kiírása

s szöveg

x,y kiírás kezdetének koordi-
nátái

m karakterkészlet.

Színek megadása

FCOL f

Keretszin megadása

f szinkód.

BCOL f

Háttérszin megadása

f szinkód.

SCOL

Rajzolósi szín megadása.

PCOL

Pont szín megadása.

Szellemgrafika

SREAD s

Szellem adatainak beolvasása
s karakter típusu változó.

SDEFINE

Szellem paramétereinek értékadás.

SMODE s,m,f

Mód megadása

s szellem sorszáma
m nagyítás és prioritás
f szellem színe.

SSET s,x,y

Megjelenítés helyének megadása

s szellem sorszáma
x,y megjelenítés helyének koordinátái.

SSET s,x1,y1 TO x2,y2,

Szellem mozgatása

s szellem sorszáma
x1,y1 kiindulási pont koordinátái
x2,y2 végpont koordinátái
p mozgatási sebesség.

SSET s TO x,y,p

Szellem mozgatása az x,y pontba

s szellem sorszáma

p mozgatási sebesség.

SWAIT s

Program futásának felfüggesztése az s szellem mozgatásának idejére.

RETURN

Megszakítás befejezése.

Egyéb utasítások

GSAVE n, "f", g

Képernyőtartalom elhelyezése külső tárolón

f állomány neve

g tároló egység száma.

GLOAD n, "f", g

Rajz betöltése és megjelenítése
paraméterek megegyeznek a GSAVE-nél elmondottakkal.

GMODE s,m

Rajzadási mód megadása

s utasítás képhez rendelése

m rajzolás finomsága.

GCLEAR

Grafikus képernyő kitörlése.

GMOVE

Grafikus ábrák mozgatása.

ROT = r

PAINT utasítással kirajzolt alakza-
tok forgatása
r forgatás szöge.

SIZE = r

PAINT utasítással kirajzolt alakza-
tok nagyítása
r nagyítás mértéke.

INVERS

Grafikus kép inverzének megjeleni-
tése.

IV. F Ü G G E L É K

C 64 AIDS parancsai

Programozást segítő parancsok

APPEND "f"

Programok összemácsolása

f másolandó program neve.

AUTO s,i

Programsorszámok automatikus generálása

s sorszámozás kezdete

i növekmény.

DELETE n,m

Programrész törlése

n törlendő első utasítás sorszáma

m törlendő utolsó utasítás sorszáma.

RENUMBER s,i,n,m

Programrész átsorszámozása

s átsorszámozandó programrész első utasításának új sorszáma

i növekmény

n programrész kezdő utasításának sorszáma

m programrész utolsó utasításának sorszáma.

REPEAT

Billentyűhöz tartozó karakterek több-

Programbelövés segítése

DUMP i

Programváltozók értékének kiírása
i típusú megadott paraméter.

EDIT /s1,s2 / (n-m,Q)

Program egy karakterláncának ki-
cserélése

s1 amit cserélünk
s2 amire cserélünk
n első utasítás sorszáma
m utolsó utasítás sorszáma
Q csere vezérlése.

FIND /s/ n,m

Karakterlánc megkeresése

s keresendő karakterlánc
n,m az EDIT-nél megadott pa-
raméterek.

TRACE W

Program végrehajtásának nyomon
követése

W opcionális paraméter.

OFF

TRACE utasítás kikapcsolása.

STEP W

Program utasításonkénti végigköve-
tése

W opcionális paraméter.

Konstansok konvertálása

DEC n1...

Decimális szám konvertálása tizenhatosba és kettesbe

ni tizes számrendszerbeli számjegy.

HEX h1...

Tizenhatos számrendszerbeli szám konvertálása tizesbe és kettesbe

hi tizenhatos számrendszerbeli számjegy.

Egyéb parancsok

KILL

Rendszer kikapcsolása.

OLD

NEW paranccsal eltörölt programok visszaállítása.

Tartalomjegyzék

	oldal
Előszó	
HELP PLUS	
Bevezetés	1
1. Programozást, listázást segítő parancsok	2
1.1. Automatikus sorszámkiírás	2
1.2. Szabad memóriahely kérése	3
1.3. Program formázott listázása	3
1.4. Program átsorszámozása	4
1.5. Program sűritése	5
1.6. Programok összemácsolása	6
1.7. Egy programrészlet megkeresése	6
1.8. Programsorok törlése	7
2. Programbelővést segítő parancsok	8
2.1. Programhiba megadása megszakítás esetén	8
2.2. Program futásának nyomonkövetése	9
2.2.1. TRACE utasítás	9
2.2.2. Programvégrehajtás soronként	9
2.3. Illegális hivatkozási helyek	10
2.4. Programváltozók kiírása	10
2.4.1. Egyszerű változók értékeinek kiírása	10
2.4.2. Tömbök tartalmának kiírása	11
2.5. Konstansok konvertálása	11
2.5.1. Konvertálás tízenhatosból tízesbe	12
2.5.2. Konvertálás tízesből tízenhatosba	12

	oldal
3. Egyéb parancsok	14
3.1. Törölt program aktivizálása	14
3.2. CMD utasítás	14
3.3. Egységszám megváltoztatása	15
3.4. HELP PLUS befejezése	15
4. Disassembler parancsok	16
4.1. Disassembler aktivizálása	16
4.2. Cim és tartalom változtatása	16
4.2.1. Cimváltoztatás	17
4.2.2. Adattartalom változtatás	17
4.3. Listázás kijelölése	17
4.3.1. Listázás nyomtatón	17
4.3.2. Listázás képernyőn	18
4.4. Memória kijelölése	18
4.5. Listázás vezérlése	18
4.6. További lehetőségek	19
4.6.1. Gépi kódu program futtatása	19
4.6.2. Memóriaterület másolása	19
4.6.3. Disassembler kikapcsolása	19
5. DOS parancsok	20
5.1. Programbetöltés	20
5.2. Programbetöltés és indítás	20
5.3. Programkiírás lemezre	20
5.4. Kiírt program ellenőrzése	21
5.5. Lemeztartalom kiírása	21

	oldal
5.6. Disk státusz kiírása	21
5.7. Parancscsatornáról olvasás	22
5.8. Parancs csatorna parancsok	22
6. Assembler fordító	23
SIMON'S BASIC	24
Bevezetés	25
1.1. Grafikus utasítások	26
1.1.1. Rajzoló utasítások	26
1.1.2. Átkapcsolás grafikus módra	28
1.1.3. Átkapcsolás színes grafikus módra	29
1.1.4. Geometriai alakzatok megjelenítése	30
1.1.4.1. Egy pont megjelenítése	30
1.1.4.2. Egyenes rajzolása	31
1.1.4.3. Téglalap rajzolása	32
1.1.4.4. Görbevonalu zárt alakzat rajzolá- sa	33
1.1.4.5. Általános alakzat kirajzolása	37
1.1.5. Alakzatok színezése	40
1.1.5.1. Zárt alakzat színezése	40
1.1.5.2. Színes téglalap rajzolása	41
1.1.6. Rajzok feliratozása	42
1.1.6.1. Karakterek megjelenítése	42
1.1.6.2. Karakterláncok kiírása	43
1.1.6.3. Mintaprogram	44

	oldal
1.1.7. További grafikus utasítások	45
1.1.7.1. CSET utasítás	45
1.1.7.2. Egy pont állapotának lekérdezése	46
1.2. Hagyományos utasítások kiterjesztése	47
1.2.1. Feltételes utasítás kiterjesztése	47
1.2.1.1. IF...THEN...ELSE	47
1.2.1.2. RCOMP utasítás	47
1.2.2. Ciklusutasítás kiterjesztése	48
1.2.2.1. REPEAT...UNTIL utasítás	48
1.2.2.2. LOOP...EXIT IF...END LOOP.	49
1.2.3. Zárt szubrutin	50
1.2.4. Címkére ugrás	51
1.2.5. Helyi és globális változók	53
1.2.6. Számított ugró utasítás	54
1.2.7. DATA pointer állítása	55
1.2.8. Olvasással-írással kapcsolatos műveletek...	56
1.2.8.1. Szöveg kiírása középre	56
1.2.8.2. Kiíratási kép szerkesztése	56
1.2.8.3. Kiíratás helyének megadása	58
1.2.8.4. Bemenő adatok ellenőrzése	59
1.2.8.5. Funkció billentyű használata ...	60
1.3. Egyéb utasítások	61
1.3.1. Programfutás felfüggesztése	61
1.3.2. Hibák lekezelése programból	62
1.3.3. Programok titkosítása	64

	oldal
2. BASIC függvények kiterjesztése	66
2.1. Matematikai függvények	66
2.1.1. Modulus függvény	66
2.1.2. Egészrész	66
2.1.3. Tizedesrész függvény	67
2.2. Logikai műveletek kiterjesztése	67
2.3. Karakter műveletek	68
2.3.1. Beszúrás karaktersorozatba	68
2.3.2. Karakterlánc egy részének kicserélése ...	69
2.3.3. Karakterlánc helyének megkeresése	70
2.3.4. Karakterlánc többszörözése	71
2.4. Számkonverziós utasítások	72
2.4.1. Bináris konstans konvertálása	72
2.4.2. Hexadecimális szám konvertálása decimális- ra	72
2.5. Fényceruzával, paddle-vel és a botkormánnyal kap- csolatos függvények	72
2.5.1. Fényceruza koordinátái	72
2.5.2. Paddle potenciométerének állása	73
2.5.3. Botkormány helyzete	73
3. Képernyő műveletek	74
3.1. Képernyőn szín villogtatás	74
3.2. Keretszín villogtatása	74
3.3. Terület feltöltése karakterekkel	75
3.4. Egy terület színének megválasztása	76
3.5. Terület feltöltése színes karakterekkel	77

	oldal
3.6. Egy terület átmozgatása	77
3.7. Karakter inverz képének kiírása	78
3.8. Képernyő egy területének rollozása	79
3.9. Képernyőtartalom kiírása	81
3.9.1. Képernyőtartalom kiírása háttértárolóra ..	81
3.9.2. Képernyőtartalom beolvasása háttértároló- ról	81
3.9.3. Képernyőtartalom kiírása nyomtatón ...	82
3.10. Kurzor helyének meghatározása	82
4. Programírást segítő parancsok	84
4.1. Automatikus sorszámozás	84
4.2. Program átszámozása	85
4.3. Programok összemásolása	86
4.4. Programlistázás vezérlése	86
4.4.1. Listázás oldalanként	86
4.4.2. Listázás lassítása	87
4.5. SIMON'S utasítások kiemelése	88
4.6. Karakterlánc megkeresése	88
4.7. Programfutás nyomonkövetése	90
4.8. Változók tartalmának kiírása	91
5. Saját grafikus alakzatok készítése	92
5.1. Sprite manipuláció	92
5.1.1. Helyfoglalás a memóriában	92
5.1.2. Szellem megrajzolása	93
5.1.3. Színes sprite színeinek megadása	94

	oldal
5.1.4. Szellem felépítése	94
5.1.5. Szellem megjelenítése és mozgatása	95
5.1.6. Alakzatok összeütközésének érzékelése ...	96
5.1.7. Szellem kitörlése	98
5.2. Saját karakterek készítése	98
5.2.1. Memória átmásolása	98
5.2.2. Kicserélendő karakter megadása	99
5.3. Példa MOB szerkesztésére, mozgatására, színezésére	101
6. Zenei utasítások	102
6.1. Hangerő szabályozása	102
6.2. Hullámforma és egyéb paraméterek	102
6.3. Burkoló görbe megadása	103
6.4. Lejátszandó zene megszerkesztése	104
6.5. Zene lejátszása	105
7. Egyéb parancsok	107
7.1. Disk műveletek	107
7.2. Lemezállományok neveinek listázása	107
7.3. NEW parancs hatástalanítása	108
7.4. Funkció billentyűk használata	109
 SUPERGRAPHIK 64	 112
 Bevezetés	 113
1. Szalagos változat	114
1.1. Rendszer betöltése	114
1.2. Rajzolósi mód megadása	114

	oldal
1.3. Grafikus képernyő törlése	116
1.4. Geometriai alakzatok megjelenítése	116
1.4.1. Egy pont megjelenítése, törlése	116
1.4.2. Egyenes szakasz kirajzolása és törlése ...	117
1.4.3. Téglalap kirajzolása, eltörlése	119
1.4.4. Szabályos görbevonalu alakzatok rajzolása	121
1.5. Színezési utasítások	123
1.5.1. Háttérszín megadása	123
1.5.2. Rajzolósi szín megadása	124
1.6. Szellem grafikai utasítások	125
1.7. Egyéb utasítások	127
1.7.1. Képernyő tartalom tárolása és visszaolvasá- sa	127
1.7.2. Képernyő tartalom kiírása nyomtatón ...	129
1.7.3. Egy képernyő pont ellenőrzése	130
1.7.4. Képernyőtartalmak cserélése	130
1.7.5. Alakzatok kirajzolása	131
1.7.6. Szöveg kiírása grafikus képernyőre ...	132
1.7.7. Supergrafika kikapcsolása	133
1.8. Példa program	133
2. Lemezes változat	136
2.1. Rendszer aktivizálása	136
2.2. Zenei utasítások	137
2.2.1. Hangjellemzők megadása	137
2.2.2. Szűrő beállítása	138
2.2.3. Hang megadása és lejátszása	138
2.3. 80x50-es grafika	139
2.4. Rendszer utasításai	139
2.4.1. Rajzóli utasítások	140

	oldal
2.4.3. Input-output utasítások	141
2.4.4. Szellem grafika	142
2.4.5. Egyéb utasítások	143
 C64 AIDS	 145
1. Bevezetés	146
2. Programozást segítő parancsok	147
2.1. Programok összemácsolása	147
2.2. Programsorszámok automatikus kiírása	147
2.3. Program egyes részeinek eltörlése	147
2.4. Program újrasorszámozása	148
2.5. Karakterek sokszorozása	148
3. Programbelövés segítése	149
3.1. Változók értékének kiírása	149
3.2. Program stringjeinek kicserélése	149
3.3. String megkeresése	150
3.4. Programfutas nyomonkövetése	150
4. Konstansok konvertálása	152
5. Egyéb parancsok	153
5.1. Rendszer kikapcsolása	153
5.2. Eltörölt program visszaállítása	153
 I. SIMON'S BASIC hibaüzenetek	 154
II. HELP PLUS parancsai	156
III. SIMON'S BASIC utasításai és parancsai	162
IV. SUPERGRAPHIK 64 szalagos változat utasításai ...	181
V. SUPERGRAPHIK 64 lemezes változat utasításai ...	186
VI. C64 AIDS parancsai	192

 **commodore**

64

105 Ft.